

## 5 — Przetwarzanie wstępne. Filtracja kontekstowa.

### 5.1 Cel

- zapoznanie z pojęciem kontekstu / filtracji kontekstowej),
- zapoznanie z pojęciem konwolucji (splotu),
- zapoznanie z wybranymi metodami poprawy jakości obrazu:
  - filtry liniowe dolnoprzepustowe:
    - \* filtr uśredniający,
    - \* filtr Gaussa.
  - filtry nieliniowe:
    - \* mediana,
    - \* mediana dla obrazów kolorowych.
  - filtry liniowe górnoprzepustowe:
    - \* laplasjan,
    - \* operator Robersta, Prewitta, Sobela.
- zadanie domowe: adaptacyjna filtracja medianowa.

### 5.2 Filtry liniowe uśredniające (dolnoprzepustowe)

Jest to podstawowa rodzina filtrów stosowana w cyfrowym przetwarzaniu obrazów. Wykorzystuje się je w celu rozmazania obrazu i redukcji szumów (zakłóceń) na obrazie. Filtr określony jest przez dwa parametry: rozmiar maski (ang. *kernel*) oraz wartości współczynników maski.

Warto zwrócić uwagę, że omawiane w niniejszym rozdziale operacje generują nową wartość piksela na podstawie pewnego fragmentu obrazu (tj. kontekstu), a nie jak operacje punktowe tylko na podstawie jednego piksela.

### 5.3 Filtry nieliniowe – mediana

Filtry rozmywające redukują szum ale niekorzystnie wpływają na ostrość obrazu. Dlatego często wykorzystuje się filtry nieliniowe – np. filtr medianowy (mediana – środkowa wartość w posortowanym ciągu liczb).

Podstawowa różnica pomiędzy filtrami liniowymi, a nieliniowymi polega na tym, że przy filtracji liniowej na nową wartość piksela ma wpływ wartość wszystkich pikseli z otoczenia (uśrednianie), natomiast w przypadku filtracji nieliniowej jako nowy piksel wybierana jest któraś z wartości otoczenia – według jakiegoś wskaźnika (wartość największa, najmniejsza czy właśnie mediana).

Inne filtry nieliniowe:

- filtr modowy – moda (dominanta) zamiast mediany,
- filtr olimpijski – średnia z podzbioru otoczenia (bez wartości ekstremalnych),
- hybrydowy filtr medianowy – mediana obliczana osobno w różnych podzbiórach otoczenia (np. kształt “x”, “+”), a jako wynik brana jest mediana ze zbioru wartości elementu centralnego, mediana z “x” i mediana z “+”,

- filtr minimalny i maksymalny (będą omówione przy okazji operacji morfologicznych w dalszej części kursu).

Warto zdawać sobie sprawę, z szerokich możliwości dopasowywania rodzaju filtracji do konkretnego rozważanego problemu i rodzaju zaszumienia występującego na obrazie.

## 5.4 Filtry liniowe górnoprzepustowe (wyostrzające, wykrywające krawędzie)

Zadaniem filtrów górnoprzepustowych jest wydobywanie z obrazu składników odpowiedzialnych za szybkie zmiany jasności – konturów, krawędzi, drobnych elementów faktury.

## 5.5 Zadanie domowe: adaptacyjna filtracja medianowa

Wszystkie omówione i przetestowane powyżej filtry działały w ten sam sposób dla każdego piksela obrazu. Natomiast przy filtracji adaptacyjnej następuje zmiana parametrów filtru w zależności od statystycznych charakterystyk otoczenia rozpatrywanego piksela.

Omówiony w rozdziale 5.3 filtr medianowy dobrze eliminuje zakłócenia impulsowe o niewielkiej gęstości przestrzennej (mniej niż 20 % w pionie i poziomie). W zadaniu filtracji bardziej zakłóconych obrazów lepiej sprawdza się adaptacyjny filtr medianowy. “Adaptacyjność” filtra polega na zmianie rozmiaru okna w trakcie filtracji – w zależności od pewnych warunków.

Przyjmijmy następujące oznaczenia:

- $z_{min}$  – najmniejsza jasność w oknie  $S_{xy}$
- $z_{max}$  – największa jasność w oknie  $S_{xy}$
- $z_{med}$  – mediana z jasności w oknie  $S_{xy}$
- $z_{xy}$  – jasność w punkcie o współrzędnych  $(x, y)$
- $S_{max}$  – maksymalny dozwolony rozmiar okna  $S_{xy}$

Algorytm można opisać następującym pseudokodem:

Faza A :

$A1 = z_{med} - z_{min}$

$A2 = z_{med} - z_{max}$

**if**  $A1 > 0$  **AND**  $A2 < 0$  **then**

to przejdź od fazy B

**else**

zwiększ rozmiar okna

**if** rozmiar okna  $\leq S_{max}$  **then**

to powtórz fazę A

**else**

zwróć  $z_{med}$

**end if**

**end if**

Faza B :

$B1 = z_{xy} - z_{min}$

$B2 = z_{xy} - z_{max}$

**if**  $B1 > 0$  **AND**  $B2 < 0$  **then**

to zwróć  $z_{xy}$

**else**

zwróć  $z_{med}$

**end if**

**Wyjaśnienie działania:**

Cały czas należy pamiętać, jaki są cele działania filtru:

- usunięcie zakłóceń typu pieprz i sól (ang. *salt and pepper noise*),
- wygładzenie innego rodzaju zakłóceń,
- redukcja zniekształceń (pogrubianie albo pocienianie krawędzi).

Wartości  $z_{min}$  i  $z_{max}$  uważa się za zakłócenia impulsowe w danym oknie, nawet jeżeli nie są to największa i najmniejsza wartość w całym obrazie.

Celem fazy A jest określenie czy rezultat filtracji medianowej  $z_{med}$  jest zakłóceniem impulsowym czy nie. Jeżeli spełniona jest nierówność  $z_{min} < z_{med} < z_{max}$  wartość  $z_{med}$  nie może być zakłóceniem impulsowym. W takim przypadku przechodzimy do fazy B i sprawdzamy czy piksel  $z_{xy}$  jest zakłóceniem impulsowym. Jeżeli spełniona jest nierówność  $z_{min} < z_{xy} < z_{max}$ ,  $z_{xy}$  nie może być zakłóceniem impulsowym. W takim przypadku algorytm zwraca niezmienną wartość  $z_{xy}$  – w ten sposób zmniejsza się zniekształcenia. Jeżeli nierówność  $z_{min} < z_{xy} < z_{max}$  nie jest spełniona oznacza to, że albo  $z_{xy} = z_{min}$  albo  $z_{xy} = z_{max}$ . W obu przypadkach uznaje się, że wartość  $z_{xy}$  jest zakłóceniem i jako wynik filtracji zwraca się wartość  $z_{med}$ , o której na podstawie fazy A wiadomo, że nie jest zakłóceniem.

Przypuśćmy, że nie została spełniona nierówność  $z_{min} < z_{med} < z_{max}$ , algorytm wtedy zwiększa rozmiar okna i powtórnie wykonuje fazę A. Dzieje się tak aż do momentu, kiedy algorytm znajdzie medianę, która nie jest zakłóceniem impulsowym albo osiągnięty zostanie maksymalny rozmiar okna. W przypadku gdy osiągnięty zostanie maksymalny rozmiar okna algorytm zwraca wartość  $z_{med}$ . Nie ma gwarancji, że wartość ta nie jest zakłóceniem impulsowym, jednakże odpowiedni dobór maksymalnego rozmiaru okna pozwala zminimalizować prawdopodobieństwo wystąpienia takiej sytuacji.

Po wyznaczeniu wartości po filtracji dla danego piksela, okno filtru się przemiesza, a algorytm uruchamiany jest dla nowej lokalizacji z parametrami początkowymi.

#### Zadanie - zaimplementować metodę:

1. Utwórz nowy m-plik. Na początku wykonaj polecenia `close all; clearvars; clc;`. Wczytaj obrazy `plytkaSzumImp.bmp`. Wyświetl go.
2. Przetnij obraz filtrem medianowym o rozmiarze okna  $7 \times 7$ . Wynik filtracji wyświetl.
3. Zaimplementuj opisany adaptacyjny filtr medianowy. Wskazówki:
  - przed rozpoczęciem obliczeń należy skopiować oryginalny obraz – do tej kopii będą zapisywane wyniki filtracji
  - należy przyjąć maksymalny rozmiar okna na  $7 \times 7$ , a początkowy na  $3 \times 3$ . Zakładamy, że rozmiar filtru powinien być nieparzysty. Dla celów implementacji wygodnie jest przyjąć, że filtr  $3 \times 3$  opisuje liczba 1,  $5 \times 5 - 3$  itp.
  - obliczenia trzeba wykonać w pętli `for` (odwiedzić, każdy piksel) – dla ułatwienia zaleca się pominięcie pikseli brzegowych (jeżeli maksymalny rozmiar okna wynosi  $7 \times 7$  to nie należy dokonywać obliczeń dla 3 pikseli brzegowych)
  - w każdej iteracji należy:
    - przywrócić początkowy rozmiar okna ( $3 \times 3$ ),
    - “wyciąć” otoczenie – do nowej zmiennej przekopiować odpowiedni fragment obrazka (parametr opisujący rozmiar otoczenia należy “uzmiennić”), aby uniknąć pomyłek należy skonwertować okno do formatu `double`,
    - obliczyć parametry:  $z_{min}$ ,  $z_{max}$ ,  $z_{med}$  (funkcje `min`, `max` i `median` – warto wykorzystać składnię `A(:)` – tworzenie wektora z macierzy)
  - wykonać opisany algorytm – sposób wyboru implementacji pozostawia się dowolny – każdy jest dobry byleby działał. Uwaga: obliczenia mogą się chwilę wykonywać.
4. Porównaj rezultaty filtracji medianowej i adaptacyjnej. Wyniki zaprezentuj prowadzącemu.