ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1994
Seria: AUTOMATYKA z. 113 Nr kol. 1246

Zygmunt KUŚ
Konrad WOJCIECHOWSKI

PRZETWARZANIE MEDIANOWE JAKO SZCZEGÓLNY PRZYPADEK PRZETWARZANIA STOSOWEGO

Streszczenie. Przetwarzanie medianowe i inne operatory porządkujące mają dwie właściwości nazywane progową dekompozycją i właściwością układania w stos. Te właściwości sprawiają, że filtrację medianową można zaliczyć do szerszej klasy filtrów stosowych.

W pracy przedstawiono podstawowe właściwości filtracji medianowej i filtracji stosowej.

W pracy pokazano, że filtracja medianowa wielowartościowego sygnału do niezmiennika jest równoważna wykonaniu progowej dekompozycji sygnału do sygnału binarnego, filtracji każdego binarnego sygnału do niezmiennika binarnej filtracji medianowej i następnie wykonaniu operacji odwrotnej do progowania - sumowania sygnałów binarnych. Ta równoważność pozwala na zredukowanie analizy zastosowania filtracji medianowej do sygnałów wielopoziomowych do równoważnego problemu dla sygnałów binarnych.

MEDIAN FILTRATION AS A SPECIAL CASE OF STACK FILTRATION

Summary. The median and other rank-order operators posses two properties called the threshold decomposition and the stacking properties. This properties make that the median filter is subclass of filtering called stack filtering.

In this work basic properties of median filtration and stack filtration are presented. In this paper, it is shown that median filtering multilevel signal to is equivalent to decomposing the signal into binary signals, filtering each binary signal to a root with a binary median filter, and then reversing the decomposition. This equivalence allows problems in the analysis and the implementation of median filters for multiloevels signals to be reduced to the equivalent problems for binary signals.

МЕДИАННОЕ ПРЕОБРАЗОВАНИЕ КАК ЧАСТНЫЙ СЛУЧАЙ СТЕКОВОГО ПРЕОБРАЗОВАНИЯ

Резюме. Медианное преобразование ранговые и другие операторы имеют два свойства: пороговое разбиение и построение в стек. Из-за них можно медианную филтрацию отнести к более широкому классу стековых фильтров. В работе представлены основные свойства медианной и стековой фильтрации. Показывается, что медианная фильтрация многозначного сигнала к инвариантному признаку равноценна пороговому разбиению сигнала на бинарный сигнал, фильтрации каждого бинарного сигнала к инвариантному признаку бинарной медианной фильтрации и в конце проведению обратной опербции к пороговому ограничению - суммированию бинарных сигналов. Эта равноценность позволяет редуцировать анализ применения медианной фильтрации к многозначным сигналам к эквивалентной задаче бинарных сигналов.

1. WSTĘP

W pracy tej zostanie przedstawiona realizacja przetwarzania medianowego metodą przetwarzania stosowego oraz dwa asymetryczne filtry medianowe o ciekawej właściwości rozróżniania zakłóceń o wartości mniejszej od tła i od zakłóceń o wartości większej od tła, a także inna modyfikacja przetwarzania medianowego - rekursywna filtracja medianowa.

2. OZNACZENIA I PODSTAWOWE POJĘCIA

Na wstępie zdefiniujemy pojęcie ciągu. Funkcja $n \rightarrow a(n)$ określona na zbiorze $\{1,2,\ldots, L\}$ nosi nazwę ciągu skończonego (L-wyrazowego). W innej konwencji można ją oczywiście utożsamiać z układem uporządkowanym $(a_1, a_2,\ldots, a_L)$.

W trakcie rozważań zakładamy, że sygnał wejściowy a jest dyskretnym ciągiem o długości L, który przyjmuje wartość a(m) na pozycji m, gdzie $1 \le m \le L$ i dla każdego m ma skwantowaną wartość, która jest jedną z M-wartości: 0,1,..., M-1. Szczególnym przypadkiem takiego ciągu jest ciąg binarny, który otrzymujemy dla M=2. Dla M>2 mamy ciąg wielowartościowy, który za pomocą progowej dekompozycji można rozłożyć na M ciągów binarnych. Progową dekompozycję definiujemy następująco:

Praca finansowana z BK 603/RAu1/93/2

Definicja 2.1

Progową dekompozycją wielowartościowego ciągu **a** na poziomie **i** nazywamy ciąg, którego m-ty element ma postać:

$$t_o^i(m) = \begin{cases} 1 & \text{jeśli} \quad a(m) \geq i \\ 0 & \text{jeśli} \quad a(m) < i \end{cases} \tag{2.1}$$

gdzie: $1 \leq m \leq L$ oraz $1 \leq i \leq M\text{-}1$.

□

Ciągi binarne uzyskane w wyniku progowej dekompozycji mają tę właściwość, że posumowanie ich elementów na poszczególnych pozycjach daje ciąg wielowartościowy.

W analizie przetwarzania ciągu **a** przy przejściu przez filtr medianowy będziemy wykorzystywać również pojęcie podciągu **b** składającego się z 2N+1 ($2N+1 \leq L$) kolejnych elementów wybranych z ciągu **a**.

3. PRZETWARZANIE MEDIANOWE

Chcąc wykonać filtrację medianową dla ciągu wejściowego **a** ustalamy szerokość podciągu b (określamy N) wykorzystywanego do obliczenia wartości elementu w ciągu wyjściowym z filtru. Można powiedzieć, że przez nałożenie na ciąg wejściowy apertury o szerokości 2N+1 wybieramy z ciągu **a** kolejne podciągi **b** i na ich podstawie obliczamy kolejne wartości próbek w ciągu wyjściowym.

Wykonanie filtracji medianowej wielowartościowego ciągu **a** w standardowy sposób polega na:
- ustaleniu m,
- wybraniu z ciągu **a** podciągu **b** składającego się z następujących elementów: {m-N,...,m,...m+N} (dla m=1 elementy o indeksach mniejszych od 1 są równe pierwszemu elementowi, a dla m=L elementy o indeksach większych od L są równe L-temu elementowi),
- uszeregowaniu wybranych elementów w porządku wzrastającym,
- wybraniu z tak uporządkowanego podciągu elementu środkowego,
- nadaniu m-temu elementowi ciągu wyjściowego wartości wybranego powyżej elementu,
- w celu przetworzenia całego ciągu podstawiamy m równe kolejno 1, 2,... L.

W wyniku przetwarzania ciągu **a** otrzymujemy ciąg wyjściowy **d**, którego m-ty element określony jest zależnością:

$$d(m) = \text{mediana}\,\{a(m\text{-}N),..,a(m),...a(m+N)\}, \tag{3.1}$$

gdzie początkowe i końcowe wartości a(m) (dla m=1 i m=L) oblicza się przez powtórzenie pierwszej i ostatniej próbki N+1 razy.

Jak widzimy, w takiej realizacji przetwarzania medianowego na pewnym etapie wykonywana jest operacja porządkowania. W dalszej części pracy zostanie pokazany sposób realizacji przetwarzania medianowego nie wymagający korzystania z algorytmów sortujących oraz pozwalający na sprzętowe wykonanie przetwarzania medianowego.

4. WPROWADZENIE DO PRZETWARZANIA STOSOWEGO

Pojęcie przetwarzanie stosowe zawiera w sobie szeroką klasę przetworników zrealizowanych za pomocą progowej dekompozycji (threshold docomposition) z uwzględnieniem właściwości układania w stos (stacking property). Konstrukcję takiego przetwornika prześledźmy na przykładzie filtru medianowego (FM) o aperturze szerokości 3, zrealizowanego techniką przetwarzania stosowego, na ciągu o następujących parametrach: I = 9, M = 4, na wejściu (rys. 4.1.).

```
2 2 0 1 2 2 3 1 1 -> [FILTR MEDIANOWY] -> 2 2 1 1 2 2 2 1 1
        ↓                                            ↑
Operacja progowania                     Dodawanie wyjść binarnych
z progiem 3, 2 i 1.                                  ↑
        ↓
0 0 0 0 0 0 1 0 0 -> [BINARNY F.MED] -> 0 0 0 0 0 0 0 0
1 1 0 0 1 1 1 0 0 -> [BINARNY F.MED] -> 1 1 0 0 1 1 1 0 0
1 1 0 1 1 1 1 1 1 -> [BINARNY F.MED] -> 1 1 1 1 1 1 1 1 1
```

Rys. 4.1. Filtracja medianowa z aperturą o szerokości 3 wykonana za pomocą progowej dekompozycji

Fig. 4.1. Window width 3 median filter by threshold decomposition

Właściwość progowej dekompozycji sprawia (co ilustruje rys. 4.1.), że przetworzenie M-wartościowego ciągu przez filtr medianowy (FM) jest **równoważne** następującej procedurze:

1) Dekompozycji M-wartościowego ciągu wejściowego w zbiór M-1 ciągów binarnych. k-ty ciąg binarny, gdzie k jest liczbą całkowita ze zbioru {1, 2, ..., M-1}, jest otrzymywany przez progowanie ciągu wejściowego progiem o wartości k. Przyjmujemy wartość 1 dla elementu ciągu większego lub równego k, i 0 dla elementu ciągu mniejszego od k. Zauważmy, że sumowanie tych M-1 ciągów binarnych daje nam oryginalny ciąg wejściowy (rys. 4.1.).
2) Filtracji każdego ciągu binarnego niezależnie, za pomocą jego własnego filtru ze statystyką porządkową (np. binarnego filtru medianowego - BFM). Wszystkie te operacje mogą być wykonane równolegle. Również każdy z filtrów, dzięki ciągom binarnym na wejściu, ma bardzo prostą realizację. Każdy z nich po prostu dodaje

bity wyczytanego podciągu (podciągu określonego przez aperturę) i porównuje wynik z zadana liczbą całkowitą r; wyprowadzając zero, jeśli suma jest mniejsza od r, a jedynkę jeśli suma jest większa lub równa r. Jeżeli szerokość apertury b jest równa 2(r-1)+1, to filtr jest filtrem medianowym.

3) Dodaniu elementów ciągów pojawiających się jednocześnie na wyjściach filtrów binarnych.

Za pomocą progowej dekompozycji z wykorzystaniem właściwości układania w stos można zrealizować różne przetworniki przez zastąpienie binarnego przetwornika porządkowego odpowiednią funkcją boolowską (np. na rys. 4.1. binarnego filtru medianowego odpowiednią funkcją boolowską).

Funkcje boolowskie, którymi chcielibyśmy zastąpić BFM z rys. 4.1., muszą posiadać właściwość układania w stos, zdefiniowaną następująco:

Definicja 4.1

O funkcji binarnej **B** n - wymiarowego argumentu mówimy, że ma właściwość układania w stos, jeżeli:

$$\bar{\mathbf{x}} \geq \bar{\mathbf{y}} \Rightarrow \mathbf{B}(\bar{\mathbf{x}}) \geq \mathbf{B}(\bar{\mathbf{y}}). \qquad (4.1)$$

□

Binarny wektor $\bar{\mathbf{x}}$ jest większy równy od binarnego wektora $\bar{\mathbf{y}}$, jeżeli na każdej pozycji, na której w $\bar{\mathbf{y}}$ występuje 1, również w $\bar{\mathbf{x}}$ występuje 1.

Warunek dodatności funkcji boolowskiej związany z zapisem jej w postaci wyrażenia boolowskiego zawiera następujące twierdzenie [1]:

Twierdzenie 4.1

Funkcja boolowska **Bn** o n - wejściach ma właściwość układania w stos wtedy i tylko wtedy, jeżeli **Bn** może być wyrażona w postaci wyrażenia boolowskiego nie zawierającego żadnych dopełnień zmiennych wejściowych.

□

Podsumowując można powiedzieć, że wszystkie operacje wykonywane w trakcie przetwarzania ciągu (obrazu) prze filtr stosowy sprowadzają się do:

a) progowania M-wartościowego ciągu wejściowego,
b) obliczania wartości funkcji boolowskiej od odpowiednich podciągów,
c) detekcji numeru najwyższego poziomu (wartości k), dla której wartość funkcji boolowskiej wynosi 1,
d) przyjęciu k jako wartości wyjścia przetwornika stosowego.

5. REALIZACJA PRZETWARZANIA MEDIANOWEGO JAKO PRZETWARZANIA STOSOWEGO

W tej części pracy zostanie pokazane, że powatarzanie przetwarzania medianowego na obrazie wielowartościowym w celu uzyskania niezmiennika jest równoważne:

- wykonaniu progowej dekompozycji obrazu wielowartościowego,

- wielokrotnemu "przepuszczeniu" ciągów binarnych przez binarny przetwornik medianowy aż do uzyskania binarnych niezmienników,
- posumowaniu wyjść binarnych przetworników medianowych.

W wyniku tej ostatniej operacji uzyskujemy wielowartościowy niezmiennik przetwarzania medianowego.

Przyjmujemy następujące oznaczenia: wyjście binarnego przetwornika medianowego na poziomie progowania i, dla m-tej pozycji w obrazie oznaczymy przez $x^i_s(m)$. Natomiast $y_S(m)$ będzie wielowartościowym wyjściem przetwornika medianowego wykonanego jako przetwornik stosowy.

Zastosowanie standardowej filtracji medianowej do ciągu uzyskanego w wyniku progowej dekompozycji (def. 2.1) daje nam kolejny ciąg binarny:

$$x^i_s(m) = \text{mediana}\{t^i_0(m-N),\ldots t^i_0(m),\ldots,t^i_0(m+N)\} \tag{5.1}$$

gdzie początkowe i końcowe wartości $x^i_s(m)$ (dla m=1 i m=L) oblicza się analogicznie do (3.1).

Zależność ciągu binarnego, medianowo przefiltrowanego od wyjścia $y_S(m)$ filtru medianowego (2N+1) zastosowanego do oryginalnego M-wartościowego sygnału dostajemy w następujący sposób:

Lemat 5.1

Istnieje przekształcenie f (·) przyporządkowujące zbiorowi przefiltrowanych ciągów binarnych $x^i_s(m)$, $1 \le i \le M\text{-}1$ ciąg $y_S(m)$ elementów k-wartościowych taki, że $y_S(m) = f(x^i_s(m)), 1 \le i \le M\text{-}1$.

□

Dowód lematu 5.1

Niechaj I(A) będzie z definicji funkcją taką, że:

$$I(A) = \begin{cases} 1 & \text{, jeśli A jest prawdą,} \\ 0 & \text{, jeśli A jest fałszem.} \end{cases}$$

Zbadajmy ciąg będący wyjściem filtru binarnego, na którego wejście podano ciąg binarny otrzymany w wyniku sprogowania ciągu k-wartościowego na poziomie i. Dla m takiego, że $1 \le m \le L$ mamy:

$$x_s^i(m) = \text{mediana}\{t_o^i(m-N),\ldots,t_o^i(m),\ldots,t_o^i(m+N)\} = \tag{5.2}$$

$$= \begin{cases} 1 & \text{jesli} \quad \sum_{n=-N}^{N} t_o^i(m) \geq N+1 \\ 0 & \quad\quad\;\; \sum_{n=-N}^{N} t_o^i(m) < N+1 \end{cases} =$$

$$= I\left\{ \sum_{n=-N}^{N} t_o^i(m) \geq N+1 \right\} = \tag{5.3}$$

= I{co najmniej N+1 elementów w $t_o^i(m-N),\ldots,t_o^i(M+N)$ równa się 1} = I {co najmniej N+1 elementów w a(m-N),...

...,a(m+N) jest ≥ i} (5.4)

Z (5.4) widzimy bezpośrednio, że jeśli $x_s^p(m) = 1$, to na co najmniej N+1 pozycjach w aperturze wartość sygnału a(n), m-N ≤ n ≤ m+N, jest większa lub równa p. Co z kolei sprawia, że co najmniej N+1 a(n) w aperturze jest ≥ q dla każdego q ≤ p. To jest $x_s^q(m) = 1$, dla 1 ≤ q ≤ p. To daje nam pierwszą właściwość progowej dekompozycji:

Właściwość 5.1

Jeśli $x_s^p(m) = 1$, to wtedy $x_s^q(m) = 1$ dla 1 ≤ q ≤ p.

□

Możemy teraz zapisać przekształcenie x do y. Dla dowolnego m takiego, że 1 ≤ m ≤ L mamy:

$$\begin{aligned} y_S(m) &= \text{mediana}\,(a(m-N),\ldots,a(m),\ldots,a(m+N)) = \\ &= \max\,(0,\ i\text{: co najmniej } N+1\ a(\cdot) \text{ w aperturze jest} \geq i) = \\ &= \max(0,\ i\text{: } I(\text{co najmniej } N+1\ a(\cdot) \text{ w aperturze jest} \geq i) = 1) \end{aligned} \tag{5.5}$$

Połączenie (5.4) i (5.5) daje $y_S(m) = \max\{0, i\text{: } x_s^i(m) = 1\}$, a dzięki właściwości 5.1. progowej dekompozycji mamy:

$$y_s(m) = \sum_{i=1}^{k-1} x_s^i(m) = f\{x_s^i(m), \quad 1 \le i \le iM - 1\}. \tag{5.6}$$

□

Innymi słowy mówiąc, funkcja f(·) układa w stos sygnały binarne $x_s^i(m)$ jeden nad drugim zaczynając od i=1. Wartość wyjścia na pozycji m jest najwyższym poziomem na pozycji m, na którym pojawi się 1. Funkcja f(·) określona powyżej jest odwrotnością progowej dekompozycji, co pokazuje następujący lemat:

Lemat 5.2

Binarne ciągi $t_s^i(m)$, $1 \le m \le L$, oraz $1 \le i \le M\text{-}1$, otrzymane przez progowanie $y_s(m)$, $1 \le m \le L$, są dokładnie równe binarnym ciągom $x_s^i(m)$, $1 \le m \le L$, $1 \le i \le M\text{-}1$, z lematu 5.1.

Dowód lematu 5.2

Jeżeli:

$$t_s^i(m) = \begin{cases} 1 & \text{jeśli} \quad y(m) \ge i \\ 0 & \text{w p. p.} \end{cases}$$

to korzystając z (5.6) w dowodzie lematu 5.1., otrzymujemy:

$$t_s^i(m) = \begin{cases} 1 & \text{jeśli} \quad \sum_{i=q}^{k-1} x_s^q(m) \ge i \\ 0 & \text{w p.p.} \end{cases}$$

Właściwość 5.1. pozwala na redukcję powyższej zależności do następującej postaci:

$$t_s^i(m) = \begin{cases} 1 & \text{jeśli} \quad x_s^i(m) = 1 = x^i(m) \\ 0 & \text{w p.p} \end{cases} \tag{5.7}$$

□

Powyższy wynik pokazuje, że jednokrotne przejście ciągu wejściowego przez filtr medianowy jest równoważne: dekompozycji ciągu wejściowego, następnie filtracji każdego ciągu binarnego otrzymanego przez sprogowanie ciągu wejściowego na każdym poziomie i na koniec rekonstrukcji wyjścia przy użyciu f(·).

Twierdzenie 5.1

Ciąg niezmienniczy związany z FM(2N+1) można otrzymać przez progowanie oryginalnego ciągu wejściowego, wielokrotne wykonanie filtracji medianowej (2N+1) na ciągach binarnych aż do otrzymania binarnych niezmienników i wreszcie rekonstrukcje ciągu M-poziomowego z ciągów binarnych.

```
...we1 ─┐               ┌→ wy1     =   we2 ─┐              ┌→ wy2 = we3 ...
        ↓               │                   ↓              │
dekompozycja   rekonstrukcja       dekompozycja   rekonstrukcja
        │               ↑                   │              ↑
        ↓               │                   ↓              │
    binwe1          binwy1              binwe2         binwy2
a)      └→filtracja─────┘                   └→filtracja────┘

       we1 ─┐                                       ┌→ wy2
            ↓                                       │
     dekompozycja                           rekonstrukcja
            │             binwy1=binwe2             ↑
            ↓                                       │
       binwe1──→filtracja─┘   └→filtracja ──→   binwy2

b)
```

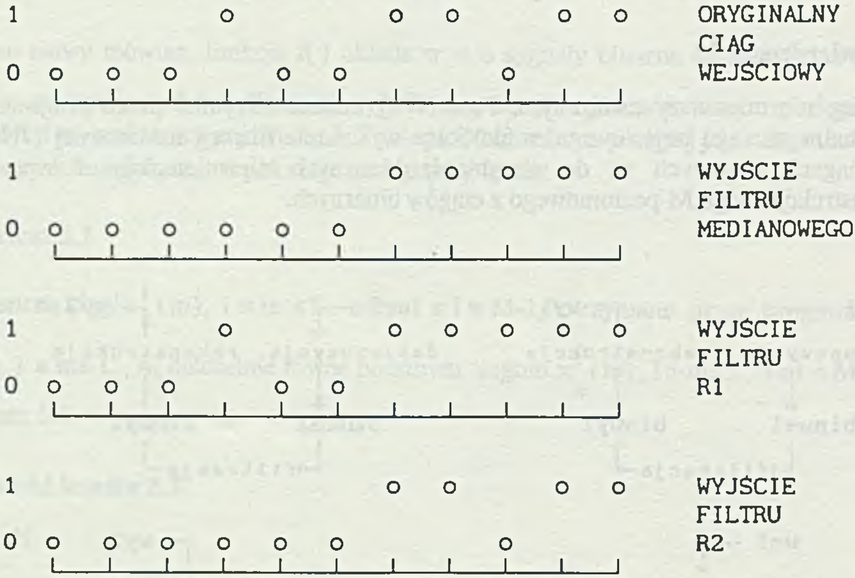
Rys. 5.1. Dwukrotne powtarzanie filtracji, a) powtarzanie dekompozycji i progowania, b) wykonanie jednokrotne dekompozycji i progowania

Fig. 5.1. Double passing by filter, a) repeated decomposition and thresholding, b) one decomposition and thresholding

Dowód powyższego twierdzenia można ograniczyć do przypomnienia, że rekonstrukcja sygnału wielowartościowego jest odwrotnością operacji progowania. Dlatego sygnały **binwy1** i **binwe2** na rys. 5.1. są sobie równe.

6. ASYMETRYCZNE FITRACJE MEDIANOWE

Zauważmy, że zbiór niezmienników filtrów bazujących na funkcjach boolowskich $x_2 + x_1 \cdot x_3$ i $x_1 \cdot x_2 + x_2 \cdot x_3$ to szczególne przypadki niezmienników filtracji medianowej. Pierwszy z nich eliminuje "impulsy ujemne", ale zachowuje "dodatnie impulsy", drugi natomiast przeciwnie. Podczas gdy filtracja medianowa (dla apertury o szerokości trzy) eliminuje "impulsy" w obu kierunkach. Z powodu tych właściwości nazywamy filtry oparte na funkcjach boolowskich $x_2 + x_1 \cdot x_3$ i $x_1 \cdot x_2 + x_2 \cdot x_3$ asymetrycznymi filtrami medianowymi (AFM). Działanie tych filtrów może nam przybliżyć rys. 6.1.

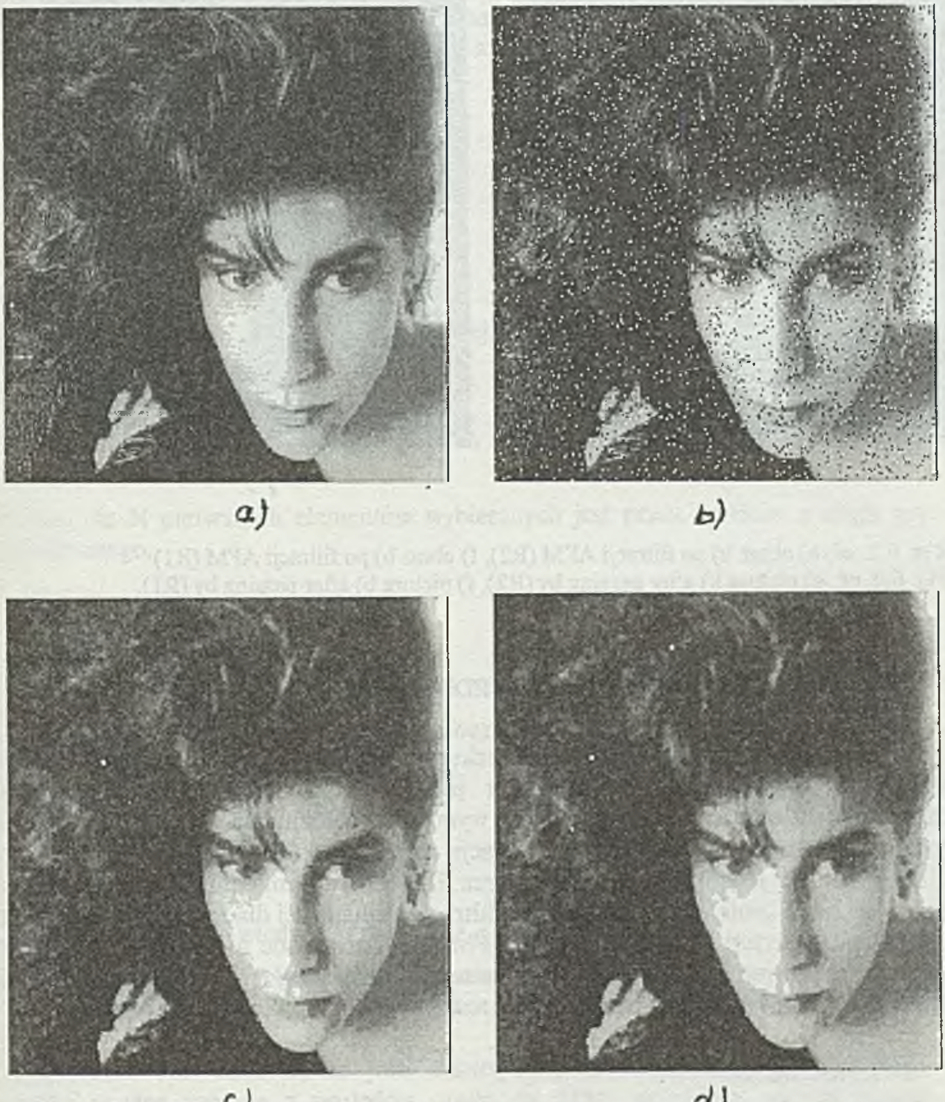


Rys. 6.1. Filtracja ciągu binarnego filtrem medianowym z aperturą o szerokości 3, asymetrycznym filtrem medianowym (R1) zachowującym "dodatnie impulsy" i asymetrycznym filtrem medianowym (R2) zachowującym ujemne impulsy

Fig. 6.1. Filtering of a binary signal with a median filter od window width 3, a positive impulse preserving asymmetric median filter (R1) and a negative impulse preserving asymetric median filter (R2).

Na rysunku 6.2. pokazano przykładowe efekty działania standardowej filtracji medianowej i asymetrycznych filtracji medianowych.

Asymetryczne filtry medianowe mogą być użyteczne w kilku dziedzinach. Po pierwsze można ich używać do eliminowania jasnych plamek na obrazie, podczas gdy reszta obrazu pozostanie nienaruszona. Ponadto, ponieważ dążą one do eliminowania impulsów jednego znaku, a zachowują impulsy przeciwnego znaku, można je stosować do opracowania geologicznych i biomedycznych sygnałów, w których dużo impulsów jednego znaku jest często znaczącymi cechami sygnału [3]. Filtry, które zachowują sygnały monotoniczne, mogłyby być użyteczne do eliminowania różnic skoków poziomów szarości w obrazach. Używając jednego z tych filtrów do filtracji wierszy obrazu do niezmienników, otrzymujemy ocenę gradientu dla każdego wiersza, która nie zawiera żadnych lokalnych detali obrazu. Poprzez odjęcie każdego niezmiennika od oryginalnego wiersza eliminujemy gradient, ale zachowujemy szczegóły obrazu.



Rys. 6.2. Przykład działania filtracji dla obrazu zaszumionego białymi i czarnymi punktami o rozkładzie równomiernym, a) obraz niezaszumiony, b) obraz a) zaszumiony, c) obraz b) po SFM (3 punkty w aperturze, 2-krotne przejście obrazu przez filtr), d) obraz b) po filtracji obydwoma asymetrycznymi filtrami medianowymi (3 punkty w aperturze, 2 - krotne przejście obrazu przez każdy z filtrów),

Fig. 6.2. Example filtration for picture with white and black point noise a) original picture, b) picture a) and noise, c) picture b) after double passing by R(1) and R(2) (window width 3),

e)

f)

Rys. 6.2. cd. e) obraz b) po filtracji AFM (R2), f) obraz b) po filtracji AFM (R1)
Fig. 6.2. cd. e) picture b) after passing by (R2), f) picture b) after passing by (R1).

7. REKURSYWNA FILTRACJA MEDIANOWA

Teraz przejdźmy do omówienia rekursywnej filtracji medianowej (RFM) wykonanej również przy zastosowaniu progowej dekompozycji. RFM posiada odmienne właściwości niż FM, stąd dla wyraźnego odróżnienia filtracja medianowa będzie dalej nazywana standardową filtracją medianową (SFM). Istotnie różni obie filtracje fakt, że jednokrotne przejście przez RFM daje niezmiennik, może on jednak być różny od niezmiennika standardowej filtracji medianowej dla tego samego sygnału i szerokości apertury. Ponadto wyjście RFM jest uzależnione nie tylko od rozmiaru apertury, ale również od kierunku jej przesuwania przez ciąg (tym samym również od kierunku przesuwania apertury dwuwymiarowej po obrazie dwuwymiarowym).

Twierdzenie 7.1

Niezmiennik związany z RFM o aperturze szerokości 2N+1 (RFM(2N+1)) może być otrzymany prze progowanie oryginalnego ciągu, filtracji RFM wykonanej na binarnych ciągach i przekształceniu ich z powrotem w M-poziomowy sygnał, używając funkcji f(·) pokazanej w lemacie 5.1.

□

Dowód tego twierdzenia można znaleźć w [2]. W pracy ograniczymy się do przedstawienia związanych z tym dowodem wybranych właściwości RFM. W trakcie dalszych rozważań $y_r(m)$ będzie oznaczać wyjście w chwili m przetwornika stosowego realizującego RFM(2N+1) przy przesuwaniu apertury od lewej strony do prawej przez sygnał wejściowy a(m) (podobnie dla obrazów dwuwymiarowych musielibyśmy określić kierunek przebiegania apertury po obrazie oraz uporządkowanie punktów w aperturze dwuwymiarowej). Definiując binarny ciąg będący wynikiem progowania oryginalnego ciągu na poziomie i w chwili m mamy:

$$t_s^i(m) = \begin{cases} 1 & \text{jeśli} \quad a(m) \geq i \\ 0 & \text{jeśli} \quad a(m) < i \end{cases} \tag{7.1}$$

gdzie: $1 \leq m \leq L$ oraz $1 \leq i \leq M-1$.

Zastosowanie rekursywnej filtracji medianowej do tego ciągu daje nam wyjściowy ciąg binarny na poziomie i:

$$x_r^i(m) = \text{mediana}\{x_r^i(m-N),\ldots,x_r^i(m-1),t_o^i(m),\ldots,t_o^i(M+N)\} \tag{7.2}$$

Widzimy, że N pierwszych elementów wybieranych jest przez aperturę z ciągu już przetworzonego.

Właściwość 7.1

$y_r(l) = f(x_r^i(l),\ 1 \leq i \leq M\text{-}1)$ jest to pierwszy element ciągu wyjściowego zrekonstruowany według funkcji f(·) zdefiniowanej w (5.6).

□

Właściwość 7.2

Jeśli $x_r^p(m) = 1$, to wtedy $x_r^q(m) = 1$, $1 \leq q \leq p$, $1 \leq m \leq L$.

Inaczej mówiąc zachodzi właściwość układania w stos.

□

Właściwość 7.2. oznacza, że jeśli rekursywnie filtrowane ciągi binarne będą układane w stos zgodnie z wartością progu, to wtedy pojawienie się na danym poziomie jedynki oznacza, że na poziomach mniejszych od danego również będą jedynki.

Właściwość 7.3

Jeśli pierwsze m-1 pozycji może być zdekomponowanych i zrekonstruowanych przez RFM, to wtedy:

$$x_r^i(n) = 1 \quad \Leftrightarrow \quad y_r^i(n) \geq i, \qquad 1 \leq n \leq m-1,$$

$$\text{to jest wtedy, gdy } y_r^i(n) = \sum_{i=1}^{m-1} x_r^i(n) \tag{7.3}$$

dla wszystkich takich n, że $1 \leq n \leq m\text{-}1$, gdzie m jest dowolną ustalona liczbą od 2 do L. Wtedy $x_r^i(n) = 1 \Leftrightarrow y_r(n) \geq i$.

□

Właściwość 7.4

Jeśli m-1 pozycji sygnału może być zdekomponowane i odtworzone przez RFM, to w ten sam sposób może być również zdekomponowana i odtworzona m-ta pozycja.

□

LITERATURA

[1] Peter D. Wendt, Edward J. Coyle, Neal C. Gallagher: Stack Filters, IEEE Transactions On Acoustaics, Speech And Signal Processing, vol. ASS-34, pp. 898-911, August 1986.

[2] J. Patrick Fitch, Edward J. Coyle, Neal C. Gallagher: Median Filtering by Threshold Decomposition, IEEE Transactions On Acoustics, Speech And Signal Processing, vol ASSP-32, pp. 1183-1188, Decomber 1984.

[3] M. Bassseville, A. Benveniste: Desing and comparative study of some sequential jump detection algorithms for digital signals, IEEE Transactions On Acoustics, Speech And Signal Processing, vol ASSP-31, pp. 521-535, JUne 1983.

Recenzent: Doc. dr inż. Bogdan Wołczak

Wpłynęło do Redakcji 5.01.1994 r.

Abstract

The median and other rank-order operators posses two properties called the threshold decomposition and the stacking properties. This properties make that the median filter is subclass of filtering called stack filtering.

In this work basic properties of median filtration and stack filtration are presented.

Repeated application of the median filter on a signal of finite length ultimately results in a sequence, termed a root signal, which is invariant to further passes of the median filter. In this paper, it is shown that median filtering multilevel signal to its root is equivalent to decomposing the signal into binary signals, filtering each binary signal to a root with a binary median filter, and then reversing the decomposition. This equivalence allows problems in the analysis and the implementation of median filters for multilevels signals to be reduced to the equivalent problems for binary signals.

The work shows that the decomposition has an important impact on the implementation of median filters. It shows that a median filter for multilevel signals is simply a parallel conection of filters for binary signals. Furthermore, since the output of the median filter for a binary signal is found by counting the number of 1's in the window and comparing the result to a threshold, these filters become almost trivial for implement ation - complicated ranking is no longer needed. The possibility of VLSI implementation is apparent.