

Uniwersytet Rzeszowski
Wydział Nauk Ścisłych i Technicznych
Instytut Informatyki



BIOMETRYCZNE SYSTEMY ZABEZPIECZEŃ

INSTRUKCJA DO ĆWICZEŃ LABORATORYJNYCH

TREŚCI KSZTAŁCENIA: OPERACJE MORFOLOGICZNE NA OBRAZACH

Spis treści

1. Cele laboratorium.....	2
2. Wprowadzenie.....	2
3. Zadania do samodzielnego rozwiązania	7

1. Cele laboratorium

Operacje morfologiczne są kluczowym narzędziem w przetwarzaniu obrazów, szczególnie w analizie kształtów i struktury obiektów. Dzięki wykorzystaniu odpowiednich elementów strukturalnych można dostosować te operacje do różnych zadań, takich jak segmentacja, poprawa jakości obrazu czy analiza konturów. Celem laboratorium jest zapoznanie się z operacjami morfologicznymi na obrazach.

2. Wprowadzenie

Operacje morfologiczne to techniki przetwarzania obrazów stosowane głównie do analizy kształtu i struktury obiektów w obrazach binarnych oraz w pewnym stopniu w obrazach w skali szarości. Ich główną cechą jest wykorzystanie elementów strukturalnych do modyfikowania obrazu.

Szkieletyzacja - pojęcie odnoszące się do procesu ścieniania linii konturowych obiektu- ze względu na swoje możliwości znalazła zastosowanie w wielu dziedzinach. Wykorzystywana jest w medycynie przy analizie białych ciałek krwi, chromosomów oraz naczyń wieńcowych, w kryminalistyce do rozpoznawania odcisków palców, przy pomiarach zanieczyszczeń obrazów, w przemyśle elektronicznym do analizy obwodów drukowanych. Często stosowana jest w wektoryzacji, czyli zamianie obrazu rastrowego na obraz wektorowy. W tym świetle autorzy chcą przedstawić sposób działania kilku wybranych algorytmów szkieletyzujących: Arcelli-Levialdi-Pavlidis, Chin-Van-Stover-Iverson, Rosenfeld-Peleg. W tym celu zostanie zaprojektowana aplikacja pokazująca działanie tych algorytmów, która będzie umożliwiać użytkownikowi wprowadzanie dowolnych szablonów szkieletyzujących oraz obserwowanie otrzymanych wyników na ekranie.

1. Szkieletyzacja obrazów binarnych

Spośród klasycznych metod, które realizują algorytm ścieniania zwany potocznie w literaturze jako skeletoning lub też thinning algorithms, wyróżnić można trzy najbardziej charakterystyczne:

Wypełniania działów wodnych, algorytm charakterystyczny dla obrazów z odcieniami szarości, w których linia grzbietowa zawiera się w izochromie jasności. Wówczas stopniowe "zalewanie" pozwoli otrzymać mapę grzbietów, przy stosunkowo niskim koszcie obliczeniowym. Wśród nich można wyróżnić jako pewną mutację przekształcenie top-hat.

Hit or Miss Transform której przykładem są algorytmy ścieniające zarówno dla obrazów czarno-białych jak i dla obrazów szarych.

Medial Axis Transform, opisywanej jako przykład transformacji posiadającej swe przekształcenie odwrotne.

Wyżej wymienione metody wykorzystują realizację dwóch podstawowych przekształceń morfologicznych tj. erozji zwanej również wytrawianiem oraz dylatacji nazywanej rozrostem. Formalnie dla obrazów binarnych zapisalibyśmy dylatację jako:

$$A \oplus B = \bigcup_{b \in B} (\bigcup_{a \in A} (a + b)) \quad (1)$$

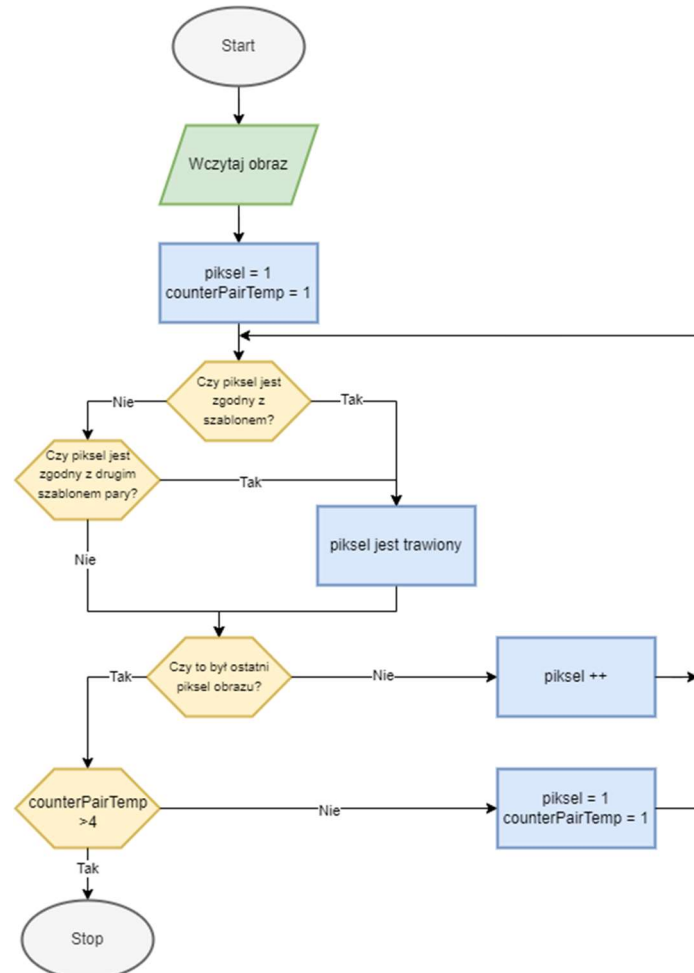
gdzie zbiory A i B oznaczają odpowiednio obraz i maskę stosowanego w przekształceniu filtru, operator \oplus oznacza sumę wektorową. Znak \bigcup oznacza algebraiczny symbol sumy w ujęciu zbiorowym. Zapis ten oznacza, że cały obraz A przeglądany jest w cyklach, których liczba odpowiada ilości poszczególnych szablonów zawartych w zbiorze B . Podobnie operację erozji zapisalibyśmy jako:

$$A \ominus B = \bigcap_{b \in B} (\bigcup_{a \in A} (a + b)) \quad (2)$$

gdzie znak \bigcap oznacza algebraiczny symbol iloczynu zbiorów. W praktyce stosowanie obu operacji składa się na tzw. przekształcenie Hit or Miss Transform. Jest to powszechnie wykorzystywana operacja morfologiczna, którą w języku naturalnym można opisać następująco:

- przyłożyć element strukturalny b do każdego punktu obrazu,
- jeśli element b nie pokrywa się z jego sąsiedztwem punkt pozostaje bez zmian,
- jeśli element b pokrywa się z sąsiedztwem dokonać zmiany wartości jasności, przeważnie na 0.

Na Rys. 1 przedstawiono schemat powyższych operacji:



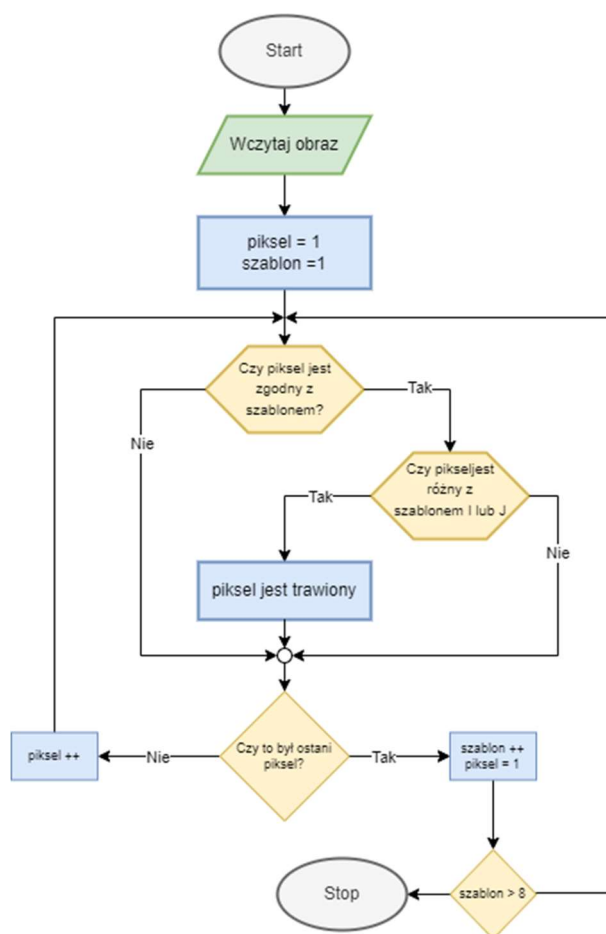
Rys. 1. Mechanizm działania algorytmu Arcelli et al.

Zwykle obraz przeglądany jest oknem, które składa się z bieżącego piksela i jego 8 punktowego otoczenia. Zdefiniowany jest również zbiór ustalonych otoczeń punktu (x, y) , który określa jakiego typu elementy składowe obrazu mają być przetworzone. I tak Arcelli i inni zaproponowali poniższy zbiór szablonów, który po pewnej modyfikacji został wykorzystany w niniejszej pracy. Ciekawy efekt podtrawiania pionowych gałęzi można uzyskać poprzez zmianę tylko dwóch pikseli w szablonach (bit 1 zamieniono na x- tzw. „bit don't care”). Na Rys. 2 zostały przedstawione szablony ze zmienionymi pikselami wyróżnionymi obwódką, a także wynik tej operacji.



Rys. 2. Szablony umożliwiające podtrawienie gałęzi pionowych i wyniki działań w wersji oryginalnej i zmodyfikowanej.

Poniżej zamieszczono inny przykład algorytmu o porównywalnej stabilności konturowej.



Rys. 3. Mechanizm działania algorytmu Iverson et al.

Algorytm Chin-Van-Stover-Iverson dokonuje porównania otoczenia piksela z szablonami 1-8, jeśli otoczenia piksela jest zgodne pierwszym szablonem, dokonywane jest porównanie z szablonem 9 oraz szablonem 10.

<input type="radio"/> wariant 1	0	0	0	0	1	x	x	1	x	x	1	0	x	0	0	0	0	x	1	x	x	1	x	x	x	x	x	0	x		
<input checked="" type="radio"/> wariant 2	1	p	1	0	p	1	1	p	1	1	p	0	1	p	0	0	p	1	0	p	1	1	p	0	0	p	1	0	x	1	x
	x	1	x	0	1	x	0	0	0	x	1	0	x	1	x	x	1	x	0	0	x	x	0	0	x	x	x	x	x	0	x



Rys. 4. Zestaw szablonów Chin-Wan-Stover-Iverson oraz otrzymany wynik szkieletyzacji obrazu binarnego.

Jeśli otoczenia piksela różni się od któregoś z tych szablonów, to piksel zostaje usunięty, w przeciwnym wypadku uznany zostaje jako fragment szkieletu. Po dokonaniu porównania wszystkich pikseli z obrazu z aktualnym szablonem, następuje zmiana szablonu na kolejny. Istnieje możliwość dowolnego wprowadzania szablonów przez użytkownika. Aby zmienić wartość piksela w szablonie trzeba użyć lewego klawisza myszki, klikając odpowiednią ilość razy, aby pojawiła się wybrana wartość, czyli 0, 1 lub x. Przy zastosowaniu dowolnych szablonów, wyniki szkieletyzacji są inne, niż przy użyciu szablonów zgodnych z algorytmami.

2. Szkieletyzacja obrazów tonowanych

Naturalnym rozwinięciem powyższych rozważań jest implementacja algorytmów szkieletyzujących obrazy szare. Wykorzystuje się tu dwie metody zaproponowane przez Rosenfeld-Peleg w wersji realizującej operację erozji oraz w wersji realizującej przekształcenie top-hat. Dla przypomnienia stosowanych przekształceń poniżej zamieszczono ich skrótowy opis. Ogólnie operację morfologiczną można określić jako pewną formę translacji sygnału $f(x)$. Wyróżnić tu można translację poziomą o wartość, którą można zapisać w poniższej formie:

$$f(x) = f(x - z) \quad (3)$$

oraz

$$(f \oplus y)(x) = f(x) + y \quad (4)$$

W przekształceniach morfologicznych często obie translacje występują łącznie według wspólnej formuły:

$$(f_z \oplus y)(x) = f(x - z) + y \quad (5)$$

Ze względu na fakt, że dziedziny sygnałów jakimi są obrazy są skończone, wygodnie jest dokonywać opisu erozji lub dyatacji jako działań na zbiorach. I tak, jeśli przyjmiemy, że $f(x) = -\infty$ dla każdego punktu spoza ramki obrazu (powiedzmy idealnie czarne tło), to dziedzinę obrazu określilibyśmy jako:

$$D[f] = \{x: f(x) > -\infty\} \quad (6)$$

Dalej powiemy, że sygnał g znajduje się „poniżej” sygnału f , jeśli dla każdego x zachodzi nierówność:

$$x \in D[g] \Rightarrow g(x) \leq f(x) \quad (7)$$

co zapiszemy jako:

$$D[g] \subseteq D[f] \quad (8)$$

Kończąc ten tok rozważań przyjmijmy oznaczenie maksimum pomiędzy sygnałami f oraz g jako $f \vee g$, zaś minimum jako $f \wedge g$. Pierwsze z nich oznaczać będzie erozję, która definiowana jest poniższą zależnością:

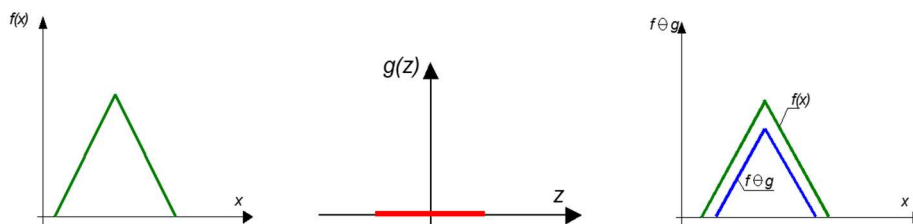
$$(f \wedge g)(x) = \begin{cases} \min\{f(x), g(x)\} & \text{if } x \in D[f] \cap D[g] \\ -\infty & \text{otherwise} \end{cases} \quad (9)$$

Przyjmując, że element strukturyzujący $g_z(x)$ jest „płaski” i równy zeru, moglibyśmy nawiązując do zapisu erozji binarnej zapisać prosto:

$$(f \ominus g)(x) = \min\{f(x) - g_z(x)\} \quad (10)$$

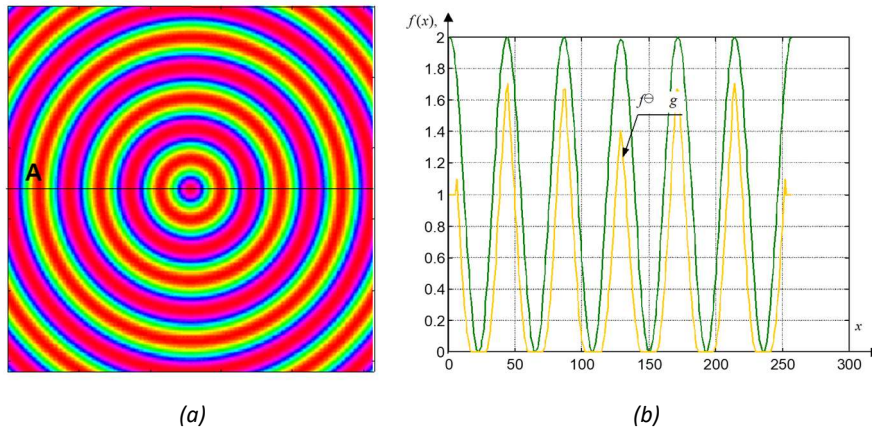
gdzie: g_z jest równe zeru dla każdego z .

W tej sytuacji otrzymamy filtr tzw. moving-minimum. Graficznie, przykład erozji sygnału f płaskim szablonem g prezentuje poniższy rysunek. Dla tak realizowanego przekształcenia można zdefiniować szablony dowolnych kształtów np. trójkątny, paraboliczny itp.



Rys. 5. Erozja sygnału f płaskim elementem strukturyzującym g .

Aby zaobserwować zachowanie się algorytmu Rosenfeld-Peleg zaprojektowano syntetyczny obraz zawierający cylindryczną falę sinusoidalną jako sygnał wejściowy dla algorytmu szkieletyzującego.



Rys. 6. Obraz syntetyczny „fala cylindryczna” (a), oraz otrzymany wynik erozji według Rosenfeld-Peleg wzdłuż przekroju A (b).

3. Zadania do samodzielnego rozwiązania

Zadanie 1.

1. Załaduj obraz testowy oraz przygotuj obraz do realizacji dalszych ćwiczeń o rozmiarach:

```
[x,map]= imread('thintest.bmp');
```

2. Wykonaj program realizujący algorytm ścieniania, którego zestaw kerneli zamieszczono poniżej:

NORTH			SOUTH			WEST			EAST		
0	0	0	1	1	X	0	X	1	X	X	0
X	1	X	X	1	X	0	1	1	1	1	0
X	1	1	0	0	0	0	X	X	1	X	0
X	0	0	X	1	X	0	0	X	X	1	X
1	1	0	0	1	1	0	1	1	1	1	0
X	1	X	0	0	X	X	1	X	X	0	0

3. Do zaprogramowania kolejnych szablonów i masek wykorzystaj poniższy fragment kodu źródłowego (funkcja `intbin8` formuje 8-bitową reprezentację liczby dziesiętnej przekazywanej parametrem `i`):

```
%wartosci prawidlowe
s=[192 80 12 5 3 65 48 20];
m=[206 87 236 117 59 93 179 213];
[f,map] = imread('thintest.bmp');

S=[];
M=[];
for i=1:8
    S(:,i)=intbin8(s(i));
    M(:,i)=intbin8(m(i));
end
```

4. Sprawdź poprawność działania algorytmu na innym obrazie testowym, dla zmodyfikowanych parametrów filtracji morfologicznej jak poniżej:

```
s=[192 2 12 5 3 65 48 20];
m=[206 58 236 117 59 93 179 213];
```

5. Wyświetl wyniki otrzymanych filtracji wraz z podpisem w oknie wykorzystując funkcję `subplot`.

6. Wyjaśnij zjawisko zmieniającej się szybkości wykonywania algorytmu w zależności od fazy jego wykonywania.

7. Wykonaj wersję algorytmu realizującego szkieletyzację wg Chin-Wan-Stover-Iverson, przyjmując zestaw kerneli jak poniżej

A	B	C	D	E	F	G	H	I	J
0	0	0	0	1	X	X	1	X	X
1	1	1	0	1	1	1	1	0	0
X	1	X	0	1	X	0	0	X	X

8. Przeprowadź dyskusję otrzymanych wyników jak dla pkt 5-6.

Wskazówka! Skrypt ułatwiający wykonanie części symulacyjnej ćwiczenia:

```
%Algorytm realizujący thinning obrazów binarnych
%=====
% VAR=1-metoda Arcelli-Cordella-Levialdi,
% VAR=2-metoda Chin-Wan-Stover-Iverson,
% S1 szablon dla metody I, M1- maska bitów dowolnych,
% S2/S3 szablon dla metody II, M2/M3- maska bitów dowolnych,

VAR= 2;    %Wariant metody

s1= [192, 80, 12, 5, 3, 65, 48, 20];
s2= [81, 69, 21, 84, 80, 65, 5, 20];
s3= [1, 64];

m1= [206, 87, 236, 117, 59, 93, 179, 213];
m2= [95, 125, 245, 215, 87, 93, 117, 213];
m3= [17, 68];

%Zamiana na postać binarną
S1=[]; S2=[]; S3=[];
M1=[]; M2=[]; M3=[];
for i=1:8
    S1(:,i)= intbin8(s1(i));
    S2(:,i)= intbin8(s2(i));
    M1(:,i)= intbin8(m1(i));
    M2(:,i)= intbin8(m2(i));
end
for i=1:2
    S3(:,i)= intbin8(s3(i));
    M3(:,i)= intbin8(m3(i));
end

[f,map]=imread('ZG.bmp');
[nr, nc]= size(f);
g=f;    %negatyw (f~=1);
%h=zeros(nr,nc); % Pojemnik na obraz przejściowy
start= tic();
c=1;    %Indeks do szablonu north, south, west, east
W= 0;    %Zmodyfikował chociaż jeden piksel 1 jeśli nie 0
W4=0;    %Licznik podcykli wdg którego przetwarzany jest obraz w określonym kierunku
l_iteracji=0 %Liczba wykonanych iteracji na obrazie do zbiegnięcia
z=0;
h=g;

while W4<5
    disp(sprintf('Iteracja= %i',l_iteracji+1))
    W=0;
    %*****
    for i=2:nr-1
        %*****
        for j=2:nc-1
            %disp(sprintf('z=%i, Iteracja=%i, Wiersz= %i, Kolumna= %i, W4= %i, W=
%i',c,l_iteracji,i,j,W4,W))
```



```

        if g(i,j)>0
            molekula= [g(i,j+1); g(i-1,j+1); g(i-1,j); g(i-1,j-1); g(i,j-1); g(i+1,j-1); g(i+1,j);
g(i+1,j+1)];
            if VAR==1 % metoda Arcelli
                for z=c:c+1 %Przykladaaj poszczegolne szablony
                    if (molekula.*M1(:,z))==S1(:,z)
                        h(i,j)=0; W=1; break;
                    end
                end
                %=====
            else %VAR==2 metoda Chin-Wan-Stover-Iverson
                if (molekula.*M2(:,8))==S2(:,8)
                    h(i,j)=0; W=1; %zmieni piksel
                else
                    if (((((molekula.*M3(:,1))==S3(:,1)) & (g(i,j+2)==0)) | ...
                        (((molekula.*M3(:,2))==S3(:,2)) & (g(i+2,j)==0))))
                        %nic nie rob
                    else
                        for z=1:7
                            if (molekula.*M2(:,z))==S2(:,z)
                                h(i,j)=0; W=1; %zmieni piksel
                                break;
                            end
                        end
                    end
                end
                %=====
            end
        end %End od if g(i,j)>0
    end %Koniec pierwszego for
    %*****

end %Koniec drugiego for
%*****
%*****

l_iteracji= l_iteracji+1;
g=h;

%Jesli metoda pierwsza to kontroluj numer cyklu
if W==0
    W4=W4+1;
else
    W4=0;
end
if (VAR==1)&(W4==4)
    break;
end
c=1+2*rem(l_iteracji,4); %Zmienia numer indeksu w tablicy filtrow
g= h;
%*****
end %Koniec While
%*****
toc(start)

```

```
imagesc((uint8(f)+2*uint8(g))/max(max(uint8(f)+uint8(g))));  
colormap(jet);
```