

Algorytmy Przetwarzania Obrazów

Operacje na obrazach (III)

WYKŁAD 4

Dla studiów stacjonarnych 2021/2023

Dr hab. Anna Korzyńska, prof. IBIB PAN

Operacje na obrazach

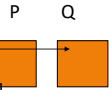
➤ Operacje punktowe (jednopunktowe):

Jednoargumentowe

$$[q(i, j)] = f[p(i, j)]$$

Wieloargumentowe

$$[q(i, j)] = f[p_1(i, j), p_2(i, j), \dots, p_k(i, j)]$$

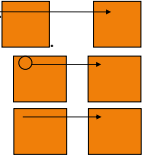


➤ Operacje sąsiedztwa (kontekstowe)

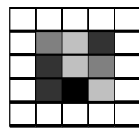
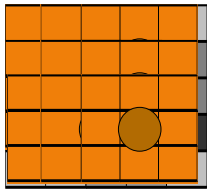
$$[q(i, j)] = f[p(i, j), p(i-1, j-1), p(i+1, j+1), \dots]$$

➤ Operacje globalne transformaty

$$[q(i, j)] = f[P]$$



Proces liczenia operacji sąsiedztwa



Do operacji sąsiedztwa E dla maski 3x3

For i=2 to X-2 do:

Begin.

For j=2 to X-2 do

Begin.

$$f_{\text{new}}(i, j) := E\{f(i-1, j-1), f(i-1, j), f(i-1, j+1), f(i, j-1), f(i, j), f(i, j+1), f(i+1, j-1), f(i+1, j), f(i+1, j+1))\}$$

End.

End.

Wynik operacji zależy od wielkości maski, ale głównie od funkcji zdefiniowanej na punkcie i jego otoczeniu.

3

Operacje morfologii matematycznej

Laboratorium nr. 6

Zadanie 1

Opracować algorytm i uruchomić aplikację realizującą operację morfologiczną na obrazie przy użyciu wybranego elementu strukturalnego. Dla operacji **erozji**, **dylatacji**, **otwarcia**, **zamknięcia** dla dwóch przypadków elementu strukturalnego:

- krzyża / romb (cztero-spójnego sąsiedztwa),
- kwadrat (ośmio-spójnego sąsiedztwa)

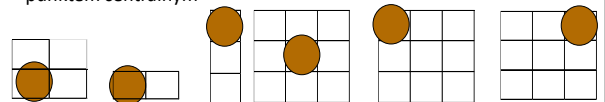
opracować algorytm i realizację programową zakładając, że będą to operacje dla obrazów binarnych o organizacji typowej dla morfologii matematycznej tzn. jasny obiekt na ciemnym tle. Zastosować jedną z 5 podanych na wykładzie metod operacji na brzegowych pikselach obrazu.

Operacje morfologii matematycznej na obrazach

Operacje pozwalające na budowanie złożonych operacji, pozwalających na analizę kształtu i wzajemnego położenia obiektów.

Fundamentalne pojęcie: **element strukturalnym (strukturujący)**

– podzbiór obrazu z wyróżnionym punktem, zwanym często punktem centralnym



6

Operacje morfologii matematycznej na obrazach

– w elemencie strukturalnym występują następujące symbole:

- 1 element wskazuje piksel zapalony tzn. wartość obiektu w masce binarnej
- 0 element wskazuje piksel wytłumiony tzn. wartość tła w masce binarnej
- X element wskazuje dowolną wartość tzn. wartość tła lub obiektu w masce binarnej

Przekształcenia polegają na **zmianie intensywności lub pozostawieniu intensywności** punktu przykrytego przez punkt centralny elementu strukturalnego w zależności od spełnienia warunków logicznych.

Operacje morfologiczne przekształcają tylko część punktów obrazu

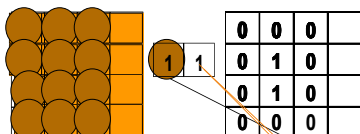
7

Operacje morfologiczne

1. Element strukturalny jest przemieszczany po wszystkich punktach obrazu tak, że punkt centralny elementu strukturalnego jest nakładany na kolejne punkty w kolejnych wierszach,
2. W każdym położeniu elementu sprawdza się, czy rzeczywista konfiguracja punktów jest **zgodna (koïncydentna)** ze wzorcem zawartym w elemencie strukturalnym zakodowanym symbolami 1, 0, X
3. W przypadku wykrycia zgodności jest wykonywana operacja związana z filtrem, a w przeciwnym przypadku wartość występująca w obrazie pierwotnym jest przepisywana.

8

Operacje morfologiczne np. dla erozji



Jeśli punkt analizowanego otoczenia punktu centralnego jest wygaszony (równy wartości tła - 0) przy zapalonym (większym od tła - 1) elemencie centralnym, wynik czyli punkt pod punktem centralnym zostaje wygaszony – ustawiony na 0,

a w przeciwnym wypadku (tzn. punkt analizowanego otoczenia jest zapalony – 1) nie wygaszamy go

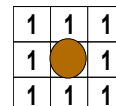
9

Podstawowe operacje morfologii matematycznej

0-zgaszony; 1-zapalony; X-o dowolnej wartości.

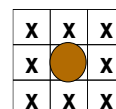
- Erozja

$$q(i, j) = \min_{i_n, j_m \in B(i, j)} (p(i_n, j_m))$$



- Dylatacja (dylacja) dualna do erozji

$$q(i, j) = \max_{i_n, j_m \in B(i, j)} (p(i_n, j_m))$$



$B(i, j)$ element strukturalny z punktem centralnym o współrzędnych (i, j)

Dylatacja jest operacją dualną do erozji i na odwrót

10

Przykłady operacji erozji



Praca dyplomowa Szymona Mireckiego

11

OpenCV

erode()

```

GMat cv::gapi::erode (
    const GMat & src,
    const Mat & kernel,
    const Point & anchor = Point(-1,-1),
    int iterations = 1,
    int borderType = BORDER_CONSTANT,
    const Scalar & borderValue = morphology
    DefaultBorderValue()
)

```

Rectangla 3x3

```

# Creating kernel
kernel = np.ones((5,5), np.uint8)

```

```

# Using cv2.erode()
method
image =
cv2.erode(image,
kernel)

```

Przykłady operacji dylacji na obrazach w skali szarości



Maksimum / Rozjaśnianie

Praca dyplomowa Szymona Mireckiego

13

OpenCV

dilate()

Rectangle 3x3

```

GMat cv::gapi::dilate(
    const GMat & src,
    const Mat & kernel,
    const Point & anchor = Point(-1,-1),
    int iterations = 1,
    int borderType = BORDER_CONSTANT,
    const Scalar & borderValue = morphologyDefaultBorderValue()
)

```

lub
zdefiniowanie
Karnel

Otwarcie

Operacja morfologiczna, która opiera się na dwóch innych operacjach: erozji i dylacji. Otwarcie polega na wykonaniu na obrazie najpierw erozji (minimum), a następnie na tak przetworzonym obrazie należy zastosować dylację (maksimum).

$$A \circ B = (A \ominus B) \oplus B$$

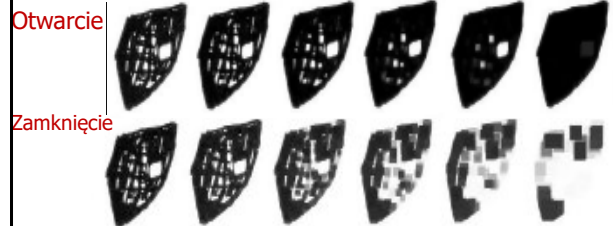
Zamknięcie (domknięcie)

Operacja morfologiczna, która opiera się na dwóch innych operacjach: dylacji i erozji. Zamknięcie polega na wykonaniu na obrazie najpierw dylacji (maksimum), a następnie na tak przetworzonym obrazie należy zastosować erozję (minimum).

$$A \bullet B = (A \oplus B) \ominus B$$

15

Operacje otwarcia i zamknięcia na obrazach szaroodcieniowych



Element strukturalny: 3x3 5x5 9x9 13x13 21x21

Komplementarność operacji

Praca dyplomowa Szymona Mireckiego

17

Operacji erozji i dylacji działających na dowolnie zdefiniowanym elemencie centralnym

morphologyEx()

```

void cv::morphologyEx (
    InputArray src,
    OutputArray dst,
    int op,
    InputArray kernel,
    Point anchor = Point(-1,-1),
    int iterations = 1,
    int borderType = BORDER_CONSTANT,
    const Scalar & borderValue = morphologyDefaultBorderValue()
)

```

getStructuringElement()

```

Mat cv::getStructuringElement (
    int shape,
    Size ksize,
    Point anchor = Point(-1,-1)
)

```

```

Python:dst=cv.morphologyEx(src, op, kernel,
    dst[, anchor[, iterations[, borderType[,
    borderValue]]]])

```

```

Python:retval=cv.getStructuringElement(
    element(shape, ksize[, anchor])
)

```

```

enum cv::MorphShapes {
    cv::MORPH_RECT = 0,
    cv::MORPH_CROSS = 1,
    cv::MORPH_ELLIPSE = 2
} shape of the structuring element More...
enum cv::MorphTypes {
    cv::MORPH_ERODE = 0,
    cv::MORPH_DILATE = 1,
    cv::MORPH_OPEN = 2,
    cv::MORPH_CLOSE = 3,
    cv::MORPH_GRADIENT = 4,
    cv::MORPH_TOPHAT = 5,
    cv::MORPH_BLACKHAT = 6,
    cv::MORPH_HITMISS = 7
} type of morphological operation More...

```

Inne operacje morfologii matematycznej

- **Otwarcie** (=Erozja+Dylacja)
- **Zamknięcie**[**domknięcie**] (=Dylacja+Erozja)
- Detekcja ekstremów Top Hat (=Zamknięcie-Obraz=Obraz-Otwarcie)
- Gradient morfologiczny (= Otwarcie+Zamknięcie)
- Wygładzanie morfologiczne (=Dylacja-Erozja)
- **Pocienianie**
- Pogrubianie
- **Szkieletyzacja** (znalezienie szkieletu czyli punktów obiektu równoodległych od jej brzegów)
- Odcinanie gałęzi (artefaktów z nieregularności obiektów szkieletyzowanych)
- Detekcja centroidów (punktów centralnych obiektu)
- Dylatacja bez styków (SKIZ ang. Skeleton by influence zone)
- Erozja warunkowa
- Rekonstrukcja (wygładzanie obszaru, czyszczenie brzegów, zalewanie dziur)
- **Automedia**

21

Operacje morfologiczne w projektach

- Operacji erozji i dylacji działających na dowolnie zdefiniowanym elemencie centralnym
- Operacje erozji warunkowej i dylacji warunkowej
- Rekonstrukcja morfologiczne przez erozję/dylację
- Operacji wyliczenia transformaty odległościowej
- Top Hat
- Ekstrakcja linii pionowych i poziomych za pomocą operacji morfologicznych
- Otoczenia wypukłe

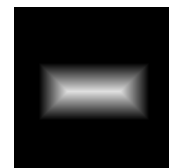
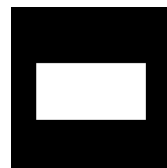
Działamy na obrazach binarnych chyba że ktoś sam rozszerzy na obrazy monochromatyczne (w szarych odcieniach)

Operacje erozji warunkowej i dylacji warunkowej

- **warunek erozji** – obiekt nie może zniknąć;
Jeśli erozja prowadzi do zniknięcia obiektu to jej wynikiem jest obraz wejściowy.
- **warunek dylacji** - obiekt nie może połączyć się z innym obiektem;
Jeśli dylacja prowadzi do połączenia z innym obiektem to wynikiem dylacji jest obraz wejściowy

Konieczność śledzenia liczby obiektów/otoczenia punktu przylącanego w ramach dylacji w poszczególnych etapach realizacji operacji

Operacji wyliczenia euklidesowej transformaty odległościowej



Obraz transformaty odległościowej obiektu, to obraz, w którym każdy piksel zawiera odległość euklidesową do najbliższego piksela brzegu obiektu

Wykonywać kolejne erozje obiektu, aż do jego zniknięcia, zapisując różnicę między obrazem wejściowym i wyjściowym dla każdego etapu (na stosie) oraz liczyć ilość elementów na stosie. Rozliczyć poziomy jasności na kolejne etapy (od najjaśniejszych na wierzchu stosu do najciemniejszych na jego spodzie) i wygenerować obraz wynikowy

Algorytmy o złożonym elemencie strukturalnym

Otoczka wypukła figury (Convex Hull; convex deficiency)

Figury wypukła na płaszczyźnie to figura dla której każda para punktów może być połączona linią prostą, która w całości przechodzi przez obszar tej figury.

Zbiór punktów na płaszczyźnie jest to wypukły jeśli tworzy figurę wypukłą.

Otoczka wypukła zbioru punktów A, to minimalny zbiór punktów XA w sąsiedztwie zbioru A, który dodany do zbioru punktów A tworzy wraz z nim zbiór punktów wypukłych.

Czyli:

jeśli A jest wypukły to $XA = \emptyset$,

jeśli nie jest wypukły to $AX \neq \emptyset$

Otoczenie wypukłe

Wypukłe otoczenie figury to najmniejsza figura wypukłą zawierającą daną figurę.

1	x	x
1	x	1
1	1	x

x	1	x
1	x	0
x	1	x



Otoczka wypukła jest liczona rekurencyjnie na $B = \{B1, B2, B3, B4\}$

Krok 0 rozpoczynając od $X_0 = A^i$
 tzn. od zbioru dla każdego i
 Krok 1 $X_k = (X_{k-1} * B^i) \cup A$
 Krok ostatni jeżeli $X_k = X_{k-1}$ to $X = X_k^i$
 Krok podsumowujący $AX = \bigcup_{i=1}^4 X_i$
 oraz minimalizacja AX – nie przekraczamy rozmiaru w pionie i poziomie

$$B1 = \begin{bmatrix} & x & x \\ x & & \\ x & & \end{bmatrix}$$

$$B2 = \begin{bmatrix} x & & x \\ x & x & x \\ x & x & x \end{bmatrix}$$

$$B3 = \begin{bmatrix} x & x & \\ x & x & x \\ x & x & \end{bmatrix}$$

$$B4 = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$$

Pocienianie

Zmniejszenie obiektu o piksele będące jego krawędzią.

$$A \ominus B = A - (A * B) = A \cap (A * \bar{B})$$

$\{B\} = \{B1, B2, \dots, Bn\}$ gdzie B_i powstaje przez rotację B_{i-1}

$$A \ominus \{B\} = ((\dots(A \ominus B1) \ominus B2)\dots) \ominus Bn$$

Pogrubianie

Zwiększenie obiektu o dodatkowe piksele w miejscu krawędzi obiektu.

$$A \odot B = A \cup (A * B)$$

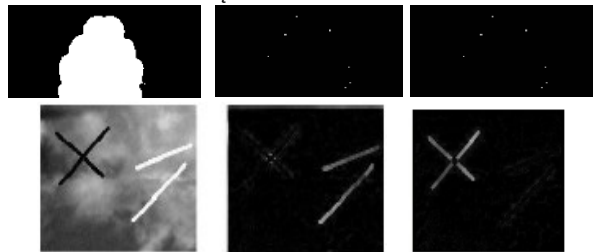
$\{B\} = \{B1, B2, \dots, Bn\}$ gdzie B_i powstaje przez rotację B_{i-1}

$$A \odot \{B\} = (((A \odot B1) \odot B2)\dots) \odot Bn$$



Top Hat

Zamknięcie - Obraz = Obraz - Otwarcie



White Top Hat

Black Top Hat

cv::MORPH_TOPHAT = 5,

30

Rekonstrukcja przez otwarcie



Otwarcie
(maska)

Rekonstrukcja
przez otwarcie

Rekonstrukcja polega na cyklicznym dokonywaniu dylatacji maski i wyznaczeniu części wspólnej z obrazu uzyskanego po dylatacji i obrazu wyjściowego przekształcenia (wykonuje się operację logiczną AND z obrazem wyjściowym. W ten sposób usuwa się te fragmenty, które zostały dodane podczas dylatacji a faktycznie będące poza odtwarzaną figurą).

31

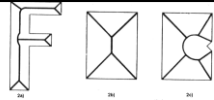
Ekstrakcja linii pionowych i poziomych za pomocą operacji morfologicznych

Operacje morfologiczne do detekcji pionowych i poziomych linii powinny mieć dobrane odpowiednie elementy strukturalne umożliwiające erozję i dylatację pozwalającą wyodrębnić takie linie:

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

1
1
1
1
1

Algorytmy szkieletyzacji



Umożliwiają upraszczanie obiektów na obrazach prowadząc do zastąpienia obiektu jego szkieletem który odzwierciedla podstawowe topologiczne własności obiektów

Jego analiza może zostać wykorzystana do:

- klasyfikacji figur ze względu na kształt,
- wyznaczania orientacji figur podłużnych,
- określania linii środkowej szerszych linii,
- rozdzielanie złączonych obiektów.

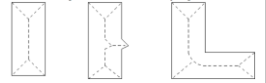
Ścienianie jest potrzebne, aby odtworzyć liniową strukturę obrazu wejściowego nie niszcząc jego spójności.

Matematyczna definicja szkieletu obiektu na płaszczyźnie ciągłej:

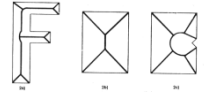
DEFINICJA 1: Niech R będzie zbiorem punktów na płaszczyźnie, B jego brzegiem, a P punktem należącym do R . Najbliższym sąsiadem punktu P na brzegu B jest punkt M należący do B taki, że nie istnieje inny punkt należący do B , którego odległość od punktu P jest mniejsza od odległości PM . Jeżeli punkt P ma więcej niż jednego najbliższego sąsiada, to P nazywamy punktem szkieletowym zbioru R . Zbiór wszystkich punktów szkieletowych jest szkieletem lub osią środkową zbioru R .

Szkieletyzacja

(ang. medial axis transformation (MAT))



Operacja, która wykrywa szkielet obiektu. Przykładowy algorytm szkieletyzacji: 1) obliczyć, ile erozji można wykonać, aby obraz nie został sprowadzony do tła, 2) wykonać obliczoną ilość razy erozję i otwarcie. Wyniki kolejnych kroków erozji i erozji z otwarciem należy od siebie odjąć. Wyniki odejmowania z kolejnych kroków należy zsumować w obraz wynikowy.



$$S(A) = \bigcup_{k=0}^K S_k(A)$$

$$S_k(A) = \bigcup_{k=0}^K \{A \ominus kB\} - [(A \ominus kB) \circ B]$$

Gdzie k to indeks kolejnej erozji, a K to maksymalna ilość erozji po którym nie dochodzi do zniknięcia obiektu

Szkielet figury, to zbiór wszystkich punktów równoodległych od co najmniej dwóch brzegów

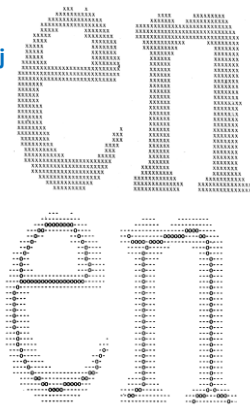
DEFINICJA 2.

Szkieletem zbioru R elementów obrazu cyfrowego jest zbiór wyznaczony w następujący sposób.

W zbiorze R określa się:

- potencjalnie szkieletowe (otoczone punktami obiektu) lub szkieletowe (stanowiące krzywą lub prostą ciągnącą się w dowolnym kierunku) oraz
- konturowe (w otoczeniu jest poziom jasności tła „0”) elementy obrazu.

Następnie usuwa się wszystkie konturowe elementy obrazu, które nie są szkieletowymi i z tak otrzymanym zbiorem R rekurencyjnie powtarzamy procedurę aż do uzyskania zbioru zawierającego jedynie szkieletowe elementy obrazu.



Rys. 4. Wynik klasycznego algorytmu ścieniania. Usunięto elementy obrazu oznaczone „-”, a szkieletowe elementy obrazu oznaczone przez „0”

Wyznaczanie szkieletu binarnego polega na wielokrotnym stosowaniu (często naprzemiennych, z różnymi elementami strukturalnymi) operacji pocieniania – do momentu, aż kolejne operacje nie wpływają na wygląd obrazu wynikowego. W tym celu można stosować różne zestawy elementów strukturalnych. Przykładem adekwatnego zestawu jest 8 elementów otrzymanych w wyniku obrotów następujących elementów strukturalnych:

$$\begin{bmatrix} 0 & 0 & 0 \\ z & 1 & z \\ 1 & 1 & 1 \end{bmatrix} \text{ oraz } \begin{bmatrix} z & 0 & 0 \\ 1 & 1 & 0 \\ z & 1 & z \end{bmatrix} \text{ o kąty } 0^\circ, 90^\circ, 180^\circ \text{ i } 270^\circ.$$

Algorytm 1 (klasyczny algorytm ścieniania):

Oznaczenia: 1 oznacza obraz wejściowy. P oznacza zbiór wzorców sąsiedztwa szkieletowych elementów obrazu wraz z obróconym o 90° pierwszym wzorcem i obróconym o 90° , 180° i 270° drugim wzorcem. Znacznik *remain* z wartością *true* wskazuje, że nieszkieletowe elementy obrazu mogą pozostać. Znacznik *skel* z wartością *true* wskazuje, że sąsiedztwo elementu obrazu odpowiada jednemu ze wzorców zbioru P . Jedynka/zero we wzorcu odpowiada niezerowemu/zerowemu elementowi w sąsiedztwie.

1. Podstaw *true* jako wartość znacznika *remain*.
2. While *remain* = *true* do kroki 3-12.
3. Podstaw *false* jako wartość znacznika *remain*. (nie dokonano żadnej zmiany)
4. For $j = 0, 2, 4$ i 6 do kroki 5-12.
5. For dla wszystkich elementów p obrazu I do kroki 6-10.
6. If $p = 1$ and if jego *j-sąsiad* = 0 then do kroki 7-10.
7. Podstaw *false* jako wartość znacznika *skel*.
8. For wszystkich wzorców P do kroki 9.
9. If sąsiedztwo p odpowiada wzorcowi P then podstaw *true* jako wartość *skel* i wyjdź z pętli.
10. If *skel* = *true* then podstaw 2 jako wartość p (szkieletowy element obrazu) else podstaw 3 jako wartość p (usuwany element obrazu) podstaw *true* jako wartość *remain*.
11. For wszystkich elementów p obrazu I do krok 12.
12. If $p = 3$, then podstaw jako p wartość 0.



$$\begin{bmatrix} 0 & 0 & 0 \\ z & 1 & z \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} z & 0 & 0 \\ 1 & 1 & 0 \\ z & 1 & z \end{bmatrix}$$

$0^\circ, 90^\circ, 180^\circ$ i 270°

13. Koniec algorytmu.

FILTRACJA DWUETAPOWA:

- 1szy etap – maska f ,
- 2gi etap (filtracja obrazu otrzymanego po 1-szym etapie) – maska g

Filtracja jednoetapowa równoważna filtracji dwuetapowej:

-maska m

Liczenie wartości elementów maski m (rozmiar 5×5) na podstawie masek f i g (rozmiary 3×3)

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	f1	f2	f3	0	0
0	0	f4	f5	f6	0	0
0	0	f7	f8	f9	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Maska f (rozmiar 3×3), współczynnik maski Kf

Maska g (rozmiar 3x3), współczynnik maski: Kg

g1	g2	g3
g4	g5	g6
g7	g8	g9

Maska m (rozmiar 5x5), Współczynnik maski: Km = Kf*Kg

m1	m2	m3	m4	m5
m6	m7	m8	m9	m10
m11	m12	m13	m14	m15
m16	m17	m18	m19	m20
m21	m22	m23	m24	m25

$$\begin{aligned}
 m1 &= 0*g1+0*g2+0*g3+0*g4+0*g5+ \\
 &\quad +0*g6+0*g7+0*g8+ f1*g9 \\
 m2 &= 0*g1+0*g2+0*g3+0*g4+0*g5+ \\
 &\quad +0*g6+0*g7+f1*g8+ f2*g9 \\
 m3 &= 0*g1+0*g2+0*g3+0*g4+0*g5+ \\
 &\quad +0*g6+ f1*g7+f2*g8+ f3*g9 \\
 &\vdots \\
 m25 &= f9*g1+0*g2+0*g3+0*g4+0*g5+ \\
 &\quad +0*g6+0*g7+0*g8+ 0*g9
 \end{aligned}$$

42

Przykładowe podstawienie:

$$\begin{aligned}
 f1=f2=f3=...=f9=1, \\
 g1=g2=g3=...=g9=1
 \end{aligned}$$

Maska g:

1	1	1
1	1	1
1	1	1

Maska wynikowa m:

1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

Współczynnik maski: Km = Kf*Kg = 1/9*1/9 = 1/81

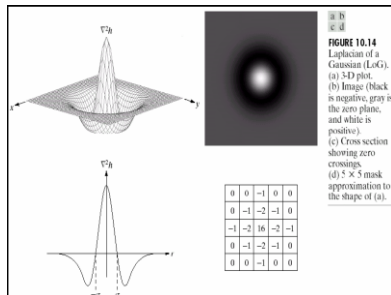
Maska f:

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

$$\begin{aligned}
 m1 &= 0*g1+0*g2+0*g3+0*g4+0*g5+ \\
 &\quad +0*g6+0*g7+0*g8+ f1*g9 \\
 m2 &= 0*g1+0*g2+0*g3+0*g4+0*g5+ \\
 &\quad +0*g6+0*g7+f1*g8+ f2*g9 \\
 m3 &= 0*g1+0*g2+0*g3+0*g4+0*g5+ \\
 &\quad +0*g6+ f1*g7+f2*g8+ f3*g9 \\
 &\vdots \\
 m25 &= f9*g1+0*g2+0*g3+0*g4+0*g5+ \\
 &\quad +0*g6+0*g7+0*g8+ 0*g9
 \end{aligned}$$

43

Laplasjan filtra gaussowskiego



Filtr jednocześnie wygładzający i wyostrzający

44

Koniec wykładu

OpenCV dla Java

- import org.opencv.core.Core;
- import org.opencv.core.CvType;
- import org.opencv.core.Mat;
- ulokowanie pliku opencv-300.jar w katalogu \opencv\build\java
- a biblioteki opencv_java3xx.dll library w katalogu: \opencv\build\java\x64 (64-bitowy system) lub \opencv\build\java\x86 (32-bitowy system).