

## Algorytmy Przetwarzania Obrazów

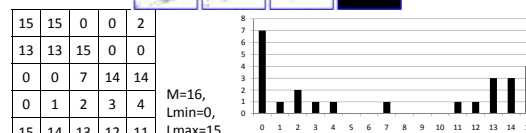
### Operacje na obrazach (I)

WYKŁAD 2

Dla studiów niestacjonarnych 2019/2020

Dr hab. Anna Korzyńska, prof. IBIB PAN

### Histogram - rozkład ilości wystąpień pikseli o zadanych poziomach jasności w obrazie



### Histogram definicja

**Histogram to wykres słupkowy przedstawiający ilość pikseli o każdej potencjalnej wartości występującej w obrazie.**

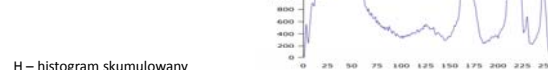
- Statystyka odzwierciedlająca rozkład jasności punktów w obrazie.
- Pewna estymata rozkładu jasności oryginalnego obrazu analogowego i rzeczywistości.



### Histogram skumulowany

h - histogram

3 7 0 10 5 6 0 5



H - histogram skumulowany

- $H(0) = h(0)$
- $H(n) = H(n-1) + h(n)$

3 10 10 20 25 31 31 36

### Algorytm tworzenia (wyznaczania wartości) histogramu

**Algorytm** wyznaczania histogramu jest prosty: analizuje każdy piksel obrazu i zlicza piksele o każdym możliwym poziomie jasności.

Oznaczenia:  $f(P)$  jest wartością elementu  $P$  w przedziale  $[0, L]$ .  $h$  jest tablicą histogramu.

Wstaw do tablicy  $h(Z)$  ( $0 \leq Z \leq L$ ) wartości zero.

For wszystkich elementów  $P$  obrazu do:

Begin.

Zwiększ  $h(f(P))$  o 1.

End.

Koniec algorytmu.

**Implementacja** jest jednak bardziej skomplikowana ze względu na różnorodność typów obrazów: monochromatyczne i kolorowe, wielo-spektralne, itp..

### Generowanie wykresu słupkowego z tablicy LUT

Wyszukaj maksymalną wartość zapisaną w tablicy  $h$

Przygotuj ramkę histogramu i współrzędne początku wykresu  $(x, y)$

For wszystkich elementów tablicy  $h$  (LUT) do:

Begin.

$h()$  unormuj wartość

Narysuj odcinek o długości odpowiadającej wartości unormowanej i punkcie zaczepienia w współrzędnych  $(x, y)$

Przesuń współrzędne  $(x, y)$

End.

Koniec algorytmu.

## Implementacja w OpenCV(C++)

```
#include <cv.h>
#include <highgui.h>

//przypisz komentarz (11b, "cv210")
//przypisz komentarz (11b, "highgui210")

//Intel Processing Library
//Intel Image
//Intel Histogram
//Intel Histogram

int main()
{
    //wczytaj obraz "XXXXXX.jpg"
    IplImage* image = cvLoadImage("XXXXXX.jpg", 1);

    //obracanie rozmiaru histogramu (ilość kolumn)
    int size[] = {256};

    //wartość kolumn i jego zmniejszona wartość
    float val;
    int norm;

    //zakres jasności
    float range[] = { 0, 256 };
    float* ranges[] = { range };

    //wartość minimalna i maksymalna histogramu
    float min_val = 0, max_val = 0;

    //konwersja wejściowego obrazu kolorowego do 8 bitowego w skali szarości
    gray = cvCreateImage( cvGetSize(image), 8, 1 );
    cvCvtColor( image, gray, CV_BGR2GRAY );

    //tworzenie okien do pokazania rezultatu
```



```
//tworzenie okien do pokazania rezultatu
cvNamedWindow("original", 1);
cvNamedWindow("gray", 1);
cvNamedWindow("histogram", 1);

//obrazy z których zostanie obliczony histogram
IplImage* images[] = { gray };

//przygotowanie pustego histogramu
hist = cvCreateHist( 1, size, CV_HIST_ANNEX, ranges, 1 );
//obliczenie histogramu z obrazu
cvCalcHist( images, hist, 0, NULL );
//zapisanie minimalnej i maksymalnej wartości histo
cvGetMinHistValue( hist, &min_val, &max_val );

//tworzenie 8-bitowego obrazu do pokazania histogramu
imgHistogram = cvCreateImage( cvSize(256, 50), 8, 1 );
//opisanie go białym kolorem
cvRectangle( imgHistogram, cvPoint(0,0), cvPoint(256,50),
CV_RGB(255,255,255), CV_FILLED );

//rysowanie histogramu jasności pikseli zmniejszonego od 0 do 50
for( int i=0; i < 256; i++ )
{
    val = cvQueryHistValue_1D( hist, i );
    norm = cvRound( val*50/max_val );
    cvLine( imgHistogram, cvPoint( i, 50 ), cvPoint( i, 50-norm ), CV_RGB(0,0,0) );
}

//pokazanie okien z rezultatem
cvShowImage( "original", image );
cvShowImage( "gray", gray );
cvShowImage( "histogram", imgHistogram );
//zapis obrazu histogramu do pliku
cvSaveImage( "histogram.jpg", imgHistogram );
cvWaitKey();

return 0;
}
```

## Operacje na histogramie

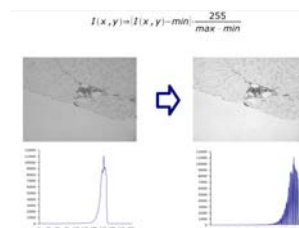
Obrazy bardzo często zawierają elementy, które są trudne do zauważenia głównie dlatego, że obiekty są mało zróżnicowane w stosunku do otoczenia.

Operacje na histogramach:

1. Rozciąganie Zadanie 1 (1)
2. Wyrównywanie Zadanie 1 (2)
3. Wyrównanie selektywne typu equalizacja

## Rozciąganie histogramu

Obrazy w których nieefektywnie wykorzystujemy pełną dynamikę odcieni i barw dostępną w danym zakresie

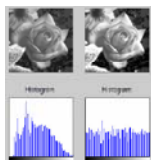


Efekt: wyostrzenie

## Wyrównywanie histogramu

Wyrównywanie histogramu:

takie przekształcenia jasności, aby wszystkie jasności były równomiernie reprezentowane w obrazie (narzędzie matematyczne to dystrybucja czyli całka z histogramu)



Wyrównanie typu equalizacja:

Rozszerzenia odległości między słupkami odpowiadającymi odcieniom szarości silnie reprezentowanymi (czyli o wysokich słupkach) a zawężenie pomiędzy słupkami o małej wysokości.

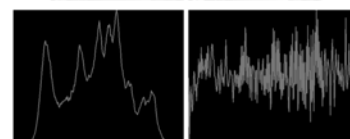
11

- Metoda ta polega na wyrównaniu poziomów szarości, tak aby histogram był płaski (czyli wszystkie stopnie szarości miały dokładnie taką samą liczbę punktów). W praktyce nie jest to możliwe, więc otrzymany histogram często jest postrzępiony.
- Do takiego wyrównania histogramu wykorzystujemy tablicę LUT.
- W pierwszym kroku musimy stworzyć dystrybucję empiryczną (odpowiednik histogramu skumulowanego)

$D[n] = (h_0 + h_1 + \dots + h_n) / \text{sum}$   
gdzie:  
hn - to ilość punktów na obrazie o n-tym poziomie szarości,  
sum - to liczba wszystkich punktów obrazu.

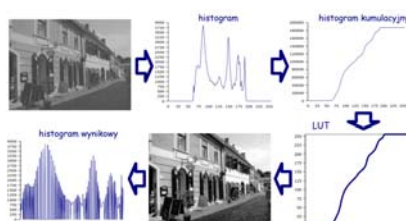
- W drugim kroku możemy stworzyć tablicę LUT:

$LUT[i] = ((D[i] - D_0) / (1 - D_0)) * (M - 1)$   
gdzie:  
D0 - to pierwsza niezerowa wartość dystrybucji obrazu źródłowego,  
M to liczba możliwych wartości jasności obrazu (zwykle 256).

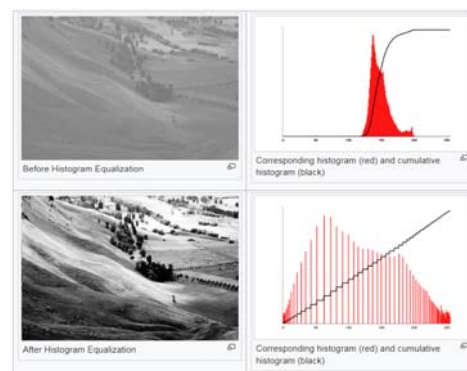


## Wyrównywanie selektywne typu equalizacja

- Wyrównywanie histogramu (*ang. Equalization*) dąży do ujednolicenia rozkładu pikseli we wszystkich fragmentach osi poziomach jasności.
- Splaszczanie – zrównanie ilości wystąpień jasności i ich rozłożenie na osi dostępnych poziomów jasności.
- Wykorzystanie całki z histogramu (dystrybucyjny) do ustalania jasności docelowych (wyjściowych).
- Proste wyrównywanie histogramu rzadko sprawia, że obraz wygląda naprawdę dobrze.



Operacja selektywnie rozciąga histogram w obszarach, gdzie występują **maksima** i selektywnie **ściąganie** (zagęszczenie prążków) histogramu w rejonach odpowiadających obszarom obrazu o słabym kontraście. Efekt: **poprawa jakości obrazu** (głównie ze względu na kontrastowość i stopień rozróżnialności szczegółów).



## Selektywne rozciąganie histogramu

Niech w obrazie pierwotnym  $p(i, j)$  poziomy szerokości przyjmują dyskretne wartości obliczane ze wzoru :

gdzie:

$L$  – liczba poziomów szerokości

$n_k$  – liczba wystąpień pikseli o tej jasności  $r_k$

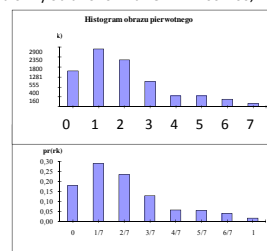
$p(r_k)$  – prawdopodobieństwo wystąpienia  $k$  – tego poziomu szerokości określane jako  $n_k / N$  gdzie  $N$  – całkowita ilość pikseli w obrazie czyli  $N1 \cdot N2$ .

Metodę wyrównywania histogramu opisuje transformanta  $T(r_k)$  odpowiadająca dystrybucji prawdopodobieństwa

$$p(s_k) = T(r_k) = \sum_{j=0}^k \frac{n_j}{N} = \sum_{j=0}^k p(r_j) \quad \begin{matrix} 0 \leq k \leq L-1 \\ k = 0, 1, 2, \dots, L-1 \end{matrix}$$

## Przykład

Załóżmy obraz o rozmiarze  $N \times N = 100 \times 100$ ,  $M=8$ .



$r_k$	$n_k$	$p(r_k)$
$r_0 = 0$	1800	0.18
$r_1 = 1/7$	2900	0.29
$r_2 = 2/7$	2350	0.24
$r_3 = 3/7$	1281	0.13
$r_4 = 4/7$	555	0.06
$r_5 = 5/7$	554	0.06
$r_6 = 6/7$	400	0.04
$r_7 = 1$	160	0.02

$$s_0 = T(r_0) = \sum_{j=0}^0 p(r_j) = p(r_0) = 0.18$$

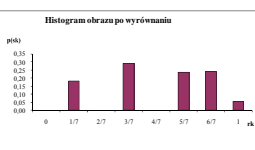
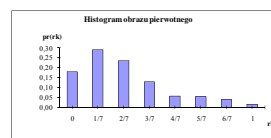
$$s_1 = T(r_1) = \sum_{j=0}^1 p(r_j) = p(r_0) + p(r_1) = 0.47$$

$$s_2 = 0.71; s_3 = 0.83; s_4 = 0.89; s_5 = 0.94; s_6 = 0.98; s_7 = 1$$

Po zaokrągleniu do możliwych poziomów szerokości:

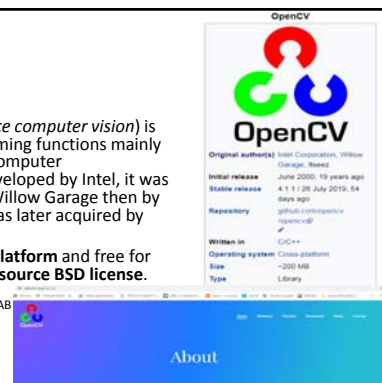
$$s_0 \approx 1/7; s_1 \approx 3/7; s_2 \approx 5/7; s_3 \approx 6/7; s_4 \approx 6/7; s_5 \approx 6/7; s_6 \approx 1; s_7 \approx 1;$$

$s_k$	$n'_k$	$p(s_k)$
0	0	0.00
1/7	1800	0.18
2/7	0	0
3/7	2900	0.29
4/7	0	0.00
5/7	2350	0.24
6/7	2390=1281+555+554	0.24
1	560=400+160	0.06



- OpenCV** (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then by firm Itseez (which was later acquired by Intel).
- The library is **cross-platform** and free for use under the **open-source BSD license**.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++.



```

* @function EqualizeHist_Demo.cpp
* @brief Demo code for equalizeHist function
* @author OpenCV team

#include "opencv2/imgcodecs.hpp"
#include "opencv2/highgui.hpp"
#include "opencv2/imgproc.hpp"
#include <iostream>
using namespace cv;
using namespace std;

/**
 * @function main
 */
int main( int argc, char** argv )
{
    //! [Load image]
    //! [Convert to grayscale]
    cvtColor( src, dst, COLOR_BGR2GRAY );
    //! [Convert to grayscale]
    //! [Apply Histogram Equalization]
    Mat dst;
    equalizeHist( src, dst );
    //! [Apply Histogram Equalization]
    //! [Display results]
    imshow( "Source image", src );
    imshow( "Equalized image", dst );
    //! [Display results]
    //! [Wait until user exits the program]
    waitKey();
    //! [Wait until user exits the program]
    return 0;

    OpenCV has a function to do
    this, cv2.equalizeHist()
}

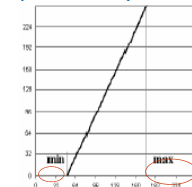
```

## Liniowe rozciąganie histogramu

Zadanie 2 (b)

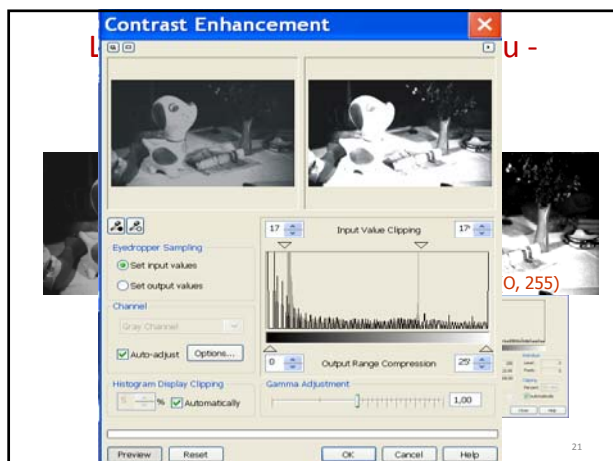
Optymalne wykorzystanie zakresu poziomów jasności:

- Zagospodarowanie całego zakresu dostępnych poziomów szarości
- Wykorzystanie skrajnych zakresów do prezentacji zakresów średnich (mało liczne, nieistotne)



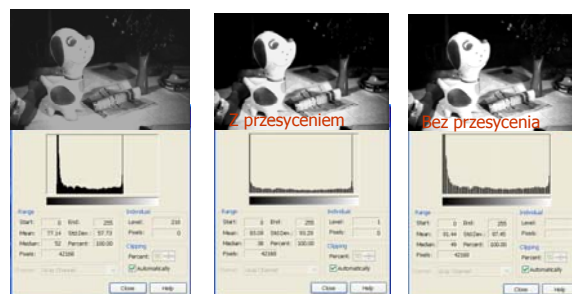
$$q(i, j) = \begin{cases} L_{\min} & \text{dla } p(i, j) < \min \\ \frac{(p(i, j) - \min) * L_{\max}}{\max - \min} & \text{dla } \min \leq p(i, j) \leq \max \\ L_{\max} & \text{dla } p(i, j) > \max \end{cases}$$

20



21

## Liniowe rozciąganie histogramu



22

## Zadanie 2 na dzisiejszym laboratorium

Oprogramowanie:

- negacji,
- progowania,
- progowania z zachowaniem poziomów szarości,
- redukcji poziomów szarości przez posteryzację,
- rozciągania z zakresu p1-p2 do zakresu q3-q4, (w szczególności gdy q3=0, q4=Lmax).

## Operacje na obrazach

Ze względu na **typy matematycznych (w tym arytmetycznych, logicznych, statystycznych) operacji**, które są wykonywane na wartościach intensywności.

Funkcje operacji na obrazach będą omawiane przy każdej operacji oddzielnie. Metody w przetwarzaniu obrazów ze względu na sposób liczenia:

- Globalne (operacje na wszystkich pikselach obrazu)
- Lokalne (operacje na ROI; ang. region of interest)
  - Punktowe;
  - Małym otoczeniu

## Operacje na obrazach

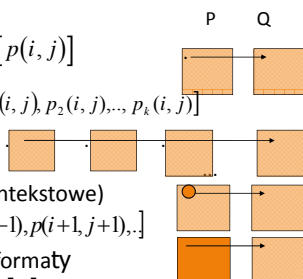
### ➤ Operacje punktowe (jednopunktowe):

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

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

## Operacje punktowe (lokalne, jednopunktowe)

26

## Operacje punktowe jednoargumentowe

Wartość piksla obrazu wyjściowego zależy od wartości piksla o takich samych współrzędnych na obrazie wejściowym (nie uwzględniamy sąsiedztwa)



Wykonywanie po wierszach, „piksel po pikselu”

27

## Operacje punktowe

Definiowane przez:

- Definicję funkcji; z jawnie postawionymi warunkami logicznymi np.: dla UOP (Uniwersalnego Operatora Punktowego);
- Wykres funkcji we współrzędnych OXY; na osi OX są **wszystkie potencjalne** wartości poziomów szarości obrazu pierwotnego P, a na OY obrazu po przekształceniu Q;
- Tablica przekodowań (LUT – Look Up Table).

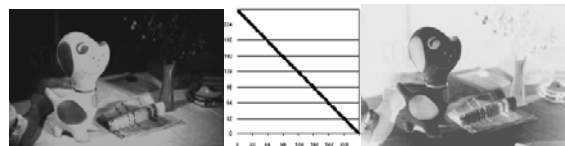
0	1	2	....	$L^P_{max}$	P
					Q

28

## Negacja

## Negacja

Negatyw obrazu



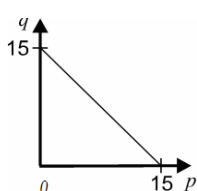
Do prezentacji informacji zawartej w ciemnych tonach (cieniach) jeśli jasne tony są nieistotne

30

### Operacja odwrotności (negacji)

$$q(i,j) = L_{\max} - p(i,j) \text{ dla } L_{\min} \leq p \leq L_{\max}$$

Dla  $L_{\min} = 0$ ,  $L_{\max} = 15$  (czyli  $M=16$ ):  $q(i,j) = 15 - p(i,j)$



[q]	0	0	15	15	13
	2	2	0	15	15
	15	15	8	1	1
	15	14	13	12	11
	0	1	2	3	4

31

### Progowanie

Jest to taka wersja operacji zmniejszenia ilości poziomów szarości do dwóch, dla której istnieje możliwość arbitralnego wyboru wartości progu ( $p_1$ ) czyli szarości granicznej, od której przyporządkowujemy wyższy poziom szarości (najczęściej biel) i poniżej której przyporządkowujemy niższy próg szarości (najczęściej czern).

$$q = \begin{cases} L_{\min} & \text{dla } p \leq p_1 \\ L_{\max} & \text{dla } p > p_1 \end{cases}$$

32

### Różne typy progowania

- Progowanie z pojedynczym progiem segmentacji

$$q = \begin{cases} L_{\min} & \text{dla } p \leq p_1 \\ L_{\max} & \text{dla } p > p_1 \end{cases}$$

- Progowanie przedziałami

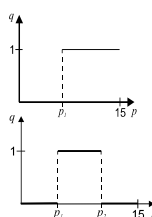
$$q = \begin{cases} L_{\max} & \text{dla } p_1 \leq p \leq p_2 \\ L_{\min} & \text{dla } p < p_1 \text{ lub } p > p_2 \end{cases}$$

- Progowanie z warunkiem spójności lub warunkiem nałożonym na wielkość obiektu

- Progowanie adaptacyjne

- Progowanie rekurencyjne

- Progowanie hierarchiczne (piramidowe, skalowalne)

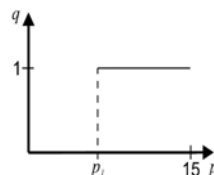


33

### Operacja progowania (binaryzacji)

$$q = \begin{cases} 0 & \text{dla } p \leq p_1 \\ 1 & \text{dla } p > p_1 \end{cases}$$

$p_1 = 5$



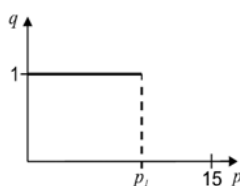
1	1	0	0	0
1	1	1	0	0
0	0	1	1	1
0	0	0	0	0
1	1	1	1	1

34

### Operacja odwrotnego progowania (binaryzacji)

$$q = \begin{cases} 1 & \text{dla } p \leq p_1 \\ 0 & \text{dla } p > p_1 \end{cases}$$

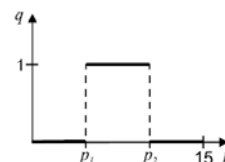
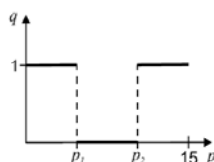
$p_1 = 5$



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

35

### Operacje progowania przedziałami (binarne)



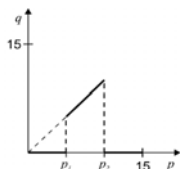
36

Operacje progowania z zachowaniem poziomów szarości

Dla  $L_{min} = 0$ ,  $L_{max} = 15$  (czyli  $M=16$ )

Z zachowaniem identyczności

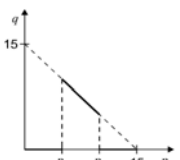
$$q = \begin{cases} p & \text{dla } p_1 \leq p \leq p_2 \\ 0 & \text{dla } p < p_1, p > p_2 \end{cases}$$



$p_1 = 2, p_2 = 12$   
Przetwarzanie obrazów |  
Wykład 3

Z zachowaniem odwrotności (negacji)

$$q = \begin{cases} L_{max} - p & \text{dla } p_1 \leq p \leq p_2 \\ 0 & \text{dla } p < p_1, p > p_2 \end{cases}$$



$p_1 = 2, p_2 = 12$

37

## Redukcja liczby poziomów szarości

Powtórna kwantyzacja na mniejszą zadaną liczbę poziomów szarości (2, 3, 4, 5, ..., 255)

Cel:

1. artystyczny – obraz postrzegany o ostrych granicach i niewielkiej ilości tonów,
2. prosta segmentacja,
3. kompresja



Inż. W. Romer  
„Izohelja” w  
Kamera Polska

38

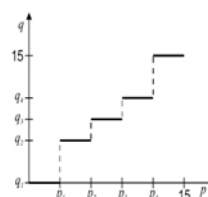
## Redukcja liczby poziomów szarości- przykłady

Różne liczby przedziałów kwantowania



## Operacja redukcji poziomów szarości

Przykład dla  $L_{min} = 0$ ,  $L_{max} = 15$  (czyli  $M=16$ )



$$q = \begin{cases} 0 & \text{dla } p \leq p_1 \\ q_1 & \text{dla } p_1 < p \leq p_2 \\ q_2 & \text{dla } p_2 < p \leq p_3 \\ q_3 & \text{dla } p_3 < p \leq p_4 \\ 15 & \text{dla } p_4 < p \leq 15 \end{cases}$$

Wyznaczyć obraz  $[q]$  i jego histogram dla:  
 $p_1 = 2, p_2 = 4, p_3 = 6, p_4 = 8; q_1 = 3, q_2 = 6, q_3 = 9;$   
Posteryzacja:  $p_2 - p_1 = p_3 - p_2 = \dots; q_2 - q_1 = q_3 - q_2 = \dots;$

Przetwarzanie obrazów |  
Wykład 3

40

## Następne laboratoria

- Operacja jednopunktowe wieloargumentowe;
- Operacje sąsiedztwa

**W operacjach jednopunktowych dwuargumentowych logicznych na obrazach działania prowadzone są na odpowiednich pikselach obrazów stanowiących argumenty danej operacji.**

**W szczególności działania prowadzone są na bitach o tej samej wadze.**

## Operacje logiczne

Poziom jasności  $n$  jest zapisany w kodzie dwójkowym jako kombinacja ośmiu 0 i 1:

Czerni 00000000

Biel 11111111

127 01000001

Operacje logiczne:

NOT NOT(1)=0; NOT(0)=1

AND 1 AND 1=1; 0 AND 0=0; 1 AND 0=0; 0 AND 1=0

OR 1 OR 1=1; 0 OR 0=0; 1 OR 0=1; 0 OR 1=1

XOR 1 XOR 1=0; 0 XOR 0=0; 1 XOR 0=1; 0 XOR 1=1

43

Przykład :

Operacja AND:  $[p] \cap [q] = [r]$

Obraz [p] :

0	0	0	1	0
1	1	15	14	0
1	0	15	13	2
2	1	0	14	0
1	0	0	0	0

Obraz [q] :

14	14	14	14	14
14	14	14	14	14
14	14	14	14	14
14	14	14	14	14
14	14	14	14	14

0000	0000	0000	0001	0000
0001	0001	1111	1110	0000
0001	0000	1111	1101	0010
0010	0001	0000	1110	0000
0001	0000	0000	0000	0000

1110	1110	1110	1110	1110
1110	1110	1110	1110	1110
1110	1110	1110	1110	1110
1110	1110	1110	1110	1110
1110	1110	1110	1110	1110

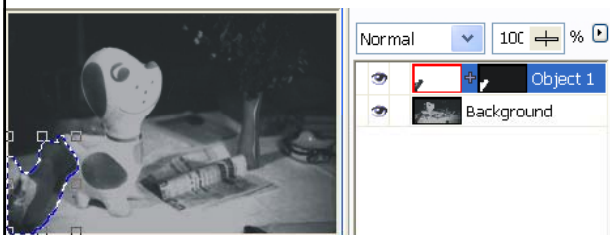
Obraz [r] :

0000	0000	0000	0000	0000
0000	0000	1110	1110	0000
0000	0000	1110	1100	0010
0010	0000	0000	1110	0000
0000	0000	0000	0000	0000

0	0	0	0	0
0	0	14	14	0
2	0	0	14	0
0	0	0	0	0

44

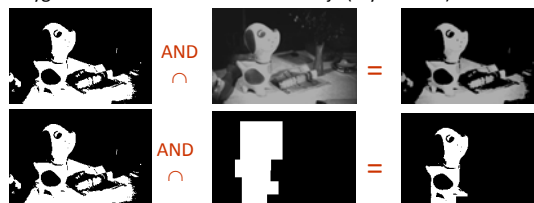
## Maska



45

## Operacja logiczna AND na obrazach

- Maskowanie, czyli selekcja fragmentów obrazów, zwanych ROI, (ang. *region of interest*) na podstawie binarnej maski
- Zacieśnianie maski
- Przygotowanie do kodowania informacji (czyszczenie)



46

## Operacja logiczna OR na obrazach

- Rozszerzanie maski
- Nakładanie informacji szyfrowanej



47

## Operacja logiczna XOR na obrazach

- Działa podobnie jak operacja OR



48



## Inne operacje punktowe

- Operacja logiczna jednoargumentowa na obrazie NOT

$$q(i, j) = \text{NOT } p(i, j)$$

Równoważna negacji

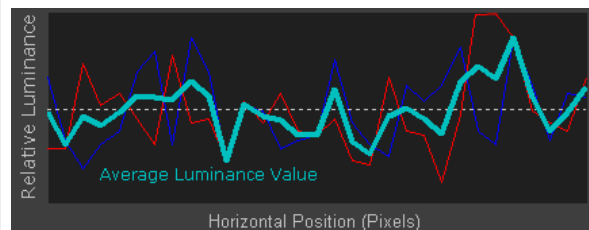
- Uśrednianie k obrazów

Stosowane do redukcji szumu (obrazy astronomiczne)

$$\left[ \begin{array}{c} \text{obraz}_1 + \text{obraz}_2 + \dots + \text{obraz}_k \end{array} \right] / n = \text{obraz}_{\text{wynikowy}}$$

$$q(i, j) = \frac{1}{k} \sum_{l=1}^k p_l(i, j)$$

49



50

## OPERACJE SĄSIEDZTWA

Są to operacje, w których na wartość zadanego piksela obrazu wynikowego  $q$  o współrzędnych  $(i, j)$  mają wpływ wartości pikseli **pewnego otoczenia** piksela obrazu pierwotnego  $p$  o współrzędnych  $(i, j)$

Przetwarzanie obrazów |  
Wykład 4

51

### Podział operacji sąsiedztwa:

- operacje wygładzania
- operacje wyostrażania

Operacje wygładzania stanowią praktyczną realizację *filtracji dolnoprzepustowej (FD)* i dzielą się na operacje filtracji **liniowej i nieliniowej**.

Operacje filtracji nieliniowej dzielą się na operacje filtracji **logicznej i medianowej**.

Operacje wyostrażania stanowią praktyczną realizację *filtracji górnoprzepustowej (FG)* i dzielą się na operacje filtracji **gradientowej i laplasjanowej**

52

## Operacje wygładzania

Podstawowe zadanie wygładzania: usuwanie zaskórczeń z obrazu

**Filtracja liniowa** (metody *konwulucyjne*, tzn. uwzględniające pewne otoczenie przetwarzanego piksela):

$$g(x, y) = \sum_{k=1}^n w_k f_k(x, y)$$

$n$  - liczba punktów (pikseli) otoczenia wraz z pikselem przetwarzanym

$f(x, y)$  - wartość piksela o współrzędnych  $x, y$  obrazu pierwotnego

$g(x, y)$  - wartość piksela o współrzędnych  $x, y$  obrazu wynikowego

$w_k$  - waga  $k$ -tego piksela otoczenia

53

### Przykład

obraz $[f(x, y)]$	otoczenie 3x3	
$[f(x, y)]$	$[f(x, y)]$	$[g(x, y)]$
14 15 13 15	15 13 15	15 13 15
12 14 0 15	14 0 15	14 11 15
13 12 12 14	12 12 14	12 12 14
15 14 14 12		

średnia ważona

$w_1$	$w_2$	$w_3$
$x-1, y-1$	$x-1, y$	$x-1, y+1$
$w_4$	$w_5$	$w_6$
$x, y-1$	$x, y$	$x, y+1$
$w_7$	$w_8$	$w_9$
$x+1, y-1$	$x+1, y$	$x+1, y+1$

$$g(x, y) = w_1 f(x-1, y-1) + w_2 f(x-1, y) + w_3 f(x-1, y+1) + w_4 f(x, y-1) + w_5 f(x, y) + w_6 f(x, y+1) + w_7 f(x+1, y-1) + w_8 f(x+1, y) + w_9 f(x+1, y+1)$$

Filtracja liniowa

Ruchoma średnia

Przetwarzanie obrazów |  
Wykład 4

54

## Sposoby zapisu operacji filtracji liniowej (wygładzanie liniowe)

1. Za pomocą macierzy wag
2. Za pomocą maski filtracji dolnoprzepustowej (FD)

## Macierz wag

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

## Maska filtracji dolnoprzepustowej

1	1	1
1	1	1
1	1	1

$$K = 1/9$$

K - współczynnik maski

55

## Przykłady macierzy wag i masek operacji filtracji liniowej

## Macierz wag

1/10	1/10	1/10
1/10	2/10	1/10
1/10	1/10	1/10

## Maska filtracji dolnoprzepustowej

1	1	1
1	2	1
1	1	1

$$K = 1/10$$

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

1	2	1
2	4	2
1	2	1

$$K = 1/16$$

56

## Metoda filtracji liniowej (wygładzanie liniowe)

$$w_k = 1/9$$

$$p_{\text{w}}(i,j) = \frac{\sum_{k=1}^9 p_k}{9}$$

$$K=1/9$$

$$q(i,j) = K \sum_{k=1}^9 p_k$$

15	15	14	13	14
14	14	13	0	12
1	1	12	13	12
0	0	1	1	10
0	0	0	0	1

[p(i,j)]

15	15	14	13	14
14	11	10	13	12
1	6	6	8	12
0	2	3	5	10
0	0	0	0	1

[q(i,j)]

57

## Metoda filtracji medianowej (wygładzanie medianowe)

p(i,j):

q(i,j):

[q(i,j)]

15	15	14	13	14
14	14	13	12	12
1	1	1	1	12
0	0	1	1	10
0	0	0	0	1

p(2,2): 1 1 12 13 14 14 14 15 15  
 p(2,3): 0 1 11 12 13 13 14 14 15  
 p(2,4): 0 11 12 12 12 13 13 14 14  
 p(3,2): 0 0 1 1 1 .....  
 p(3,3): 0 0 1 1 1 .....  
 p(3,4): 0 1 1 1 10 11 .....  
 p(4,2): 0 0 0 0 0 .....  
 p(4,3): 0 0 0 0 1 .....  
 p(4,4): 0 0 1 1 1 .....  
 p(4,5): 0 0 1 1 1 .....

58

## Filtracja medianowa (wygładzanie medianowe)

Usuwanie zakłóceń **bez rozmywania krawędzi**  
 (por. metodę filtracji liniowej)

Mediana - wartość **środkowa** (w sensie  
 położenia w ciągu wartości uporządkowanych)

59

## Operacja wygładzania medianowego

z pozostawieniem wartości skrajnych pikseli bez zmian  
 Przykładowy obraz pierwotny      Obraz wynikowy

15	15	14	13	14
14	14	13	0	12
1	1	12	13	12
0	0	1	1	10
0	0	0	0	1

15	15	14	13	14
14	14	13	12	12
1	1	1	11	12
0	0	1	1	10
0	0	0	0	1

60

### Metody operacji na pikselach wchodzących w skład skrajnych kolumn i wierszy

1. Pozostawienie wartości pikseli bez zmian
2. Wartości pikseli są nieokreślone (xxxxxxxx)
3. Nadanie pikselom wartości arbitralnie zadanych przez użytkownika (np. same wartości „0”, „15”, „10” itd.)
4. Operacje z zastosowaniem kolumn i wierszy pomocniczych (zdublowanie (powielenie) skrajnych wierszy i kolumn)
5. Operacje z wykorzystaniem pikseli z istniejącego sąsiedztwa.
  - Lewa skrajna kolumna (oprócz pikseli górnego i dolnego rogu) – kierunki 0,1,2,6,7,
  - Lewa skrajna kolumna piksel w górnym rogu – kierunki 0, 6,7,
  - Lewa skrajna kolumna (piksel w dolnym rogu) – kierunki 0,1,2,
  - Prawa skrajna kolumna (oprócz pikseli górnego i dolnego rogu) – kierunki 2,3,4,5,6,
  - Prawa skrajna kolumna piksel w górnym rogu – kierunki 4,5,6,
  - Prawa skrajna kolumna (piksel w dolnym rogu) – kierunki 2,3,4,
  - Górny skrajny wiersz (oprócz pikseli z lewego i prawego rogu) – kierunki 4,5,6,7,0
  - Dolny skrajny wiersz (oprócz pikseli z lewego i prawego rogu) – kierunki 0,1,2,3,4,

61

### Operacje wyostrozania

Metoda: konwolucja + maska *filtracji górnoprzepustowej* (FG).

W wyostrozaniu stosuje się metody numeryczne aproksymujące pochodną.

Zadanie wyostrozania:

- podkreślenie na obrazie konturów obiektów
- podkreślenie na obrazie punktów informatywnych (np. wierzchołki dla wielokątów, zakończenia, skrzyżowania, rozgałęzienia linii dla rysunków technicznych, wykresów lub pisma).

Model zadania wyostrozania: wydobywanie i uwypuklenie krawędzi obiektu.

62

### Opis matematyczny operacji wyostrozania

Model krawędzi: linia prosta separująca dwa obszary o różnej intensywności (jasności)  $I_1$  i  $I_2$ .

Użycie funkcji  $u(z)$  do matematycznego opisu krawędzi

$$u(z) = \begin{cases} 1 & \text{dla } z > 0 \\ \frac{1}{2} & \text{dla } z = 0 \\ 0 & \text{dla } z < 0 \end{cases} \quad \text{Jeśli } \delta(t) - \text{impuls Diraca, to:} \quad u(z) = \int_{-\infty}^z \delta(t) dt$$

Założenia:

Krawędź leży wzdłuż linii prostej opisanej równaniem:

$$x \sin \varphi - y \cos \varphi + \rho = 0 \quad (\text{postać normalna prostej})$$

Intensywność obrazu:

$$f(x, y) = I_1 + (I_2 - I_1) u(x \sin \varphi - y \cos \varphi + \rho)$$

63

### Wyostrozanie

Użycie cyfrowych aproksymacji gradientu i laplasjanu

$$\text{Gradient: } \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^T$$

$$\text{Moduł gradientu: } G(x, y) = \sqrt{G_x^2 + G_y^2}$$

gdzie:

$G_x, G_y$  - cyfrowe aproksymacje pochodnych.

64

### Cyfrowa wersja gradientu

Pochodna pionowa  $G_x$  funkcji  $f(x, y)$

$$G_x = \frac{def}{[f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)]}$$

maska:

	y-1	y	y+1
x-1	-1	-2	-1
x	0	0	0
x+1	1	2	1

Pochodna pozioma  $G_y$  funkcji  $f(x, y)$

$$G_y = \frac{def}{[f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)]}$$

maska:

	y-1	y	y+1
x-1	-1	0	1
x	-2	0	2
x+1	-1	0	1

$$G(x, y) = \sqrt{G_x^2 + G_y^2}$$

65

### Cyfrowa wersja laplasjanu

$$L(x, y) = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)]$$

maska:

	y-1	y	y+1
x-1	0	1	0
x	1	-4	1
x+1	0	1	0

Własności:

**Gradient:** wrażliwy na intensywność zmiany; używany tylko do detekcji krawędzi;  
**Laplasjan:** podaje dodatkową informację o położeniu piksela względem krawędzi (po jasnej czy po ciemnej stronie).

**Uwaga:** Dla operacji wyostrozania współczynnik maski  $K=1$

66

## Laplasjany



Dyskretna forma drugiej pochodnej

0	-1	0	-1	-1	-1	1	-2	1	-1	-1	-1	0	-1	0
-1	4	-1	-1	8	-1	-2	4	-2	-1	9	-1	-1	5	-1
0	-1	0	-1	-1	-1	1	-2	1	-1	-1	-1	0	-1	0

Tekst

Tekst

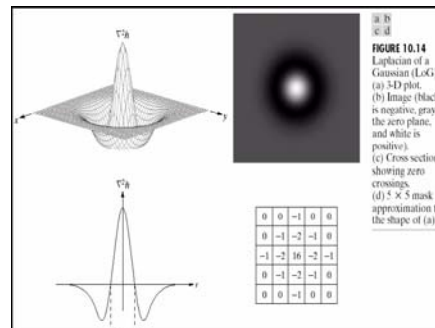
Tekst

Tekst

Tekst

67

## Laplasjan filtra gaussowskiego



68

Efekt: Obraz o wzmocnionych konturach obiektów.

Wyostrzenie - złożenie obrazów:

- wejściowego,
- po operacji zadanej laplasjanem, następnie przeskalowanie stopni szarości (jak w operacjach jednopunktowych).

a)

0	-1	0
-1	4	-1
0	-1	0

b)

-1	-1	-1
-1	8	-1
-1	-1	-1

c)

1	-2	1
-2	4	-2
1	-2	1

d)

-1	-1	-1
-1	9	-1
-1	-1	-1

e)

0	-1	0
-1	5	-1
0	-1	0

69

## Krawędź obrazu widoczna w przekroju (xz)

- obraz pierwotny
- po obróbce gradientowej (1-sza pochodna)
- po obróbce laplasjanowej (2-ga pochodna)

Krawędź

Pierwsza pochodna

$$\left(\frac{\partial f}{\partial x}\right)$$

Druga pochodna

$$\left(\frac{\partial^2 f}{\partial x^2}\right)$$

Operator gradientowy

-1	0	+1
----	---	----

Operator typu drugiej pochodnej

-1	0	+1
----	---	----

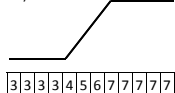
70

## Krawędź obrazu widoczna w przekroju (xz)

- obraz pierwotny
- obraz wynikowy po obróbce laplasjanowej i po dodaniu w sposób ważony jasności odpowiednich pikseli
- maska laplasjanu

Rezultat **uwypuklenie** (wzmocnianie) krawędzi (edge enhancement)

a)



b)



Wartość punktów

c)

-1	2	-1
----	---	----

71

## •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 5x5) na podstawie masek  $f$  i  $g$  (rozmiary 3x3)

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	$f_1$	$f_2$	$f_3$	0	0
0	0	$f_4$	$f_5$	$f_6$	0	0
0	0	$f_7$	$f_8$	$f_9$	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Maska  $f$  (rozmiar 3x3), współczynnik maski  $K_f$ 

72

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

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

73

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 **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

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

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

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

Wykład 5

74

## Metody skalowania tablic obrazów wynikowych

Cel skalowania: sprowadzanie wartości pikseli do zakresu  $[0, (M-1)]$

Metoda **proporcjonalna**

$$g'(x, y) = \frac{g(x, y) - g(x, y)_{\min}}{g(x, y)_{\max} - g(x, y)_{\min}} \cdot (M - 1)$$

Własność:

Równomierne przeskalowanie wszystkich pikseli obrazu.

Końcowy efekt: obraz z zakresu  $[0, (M-1)]$

75

Metoda **trójwartościowa**

$$g'(x, y) = \begin{cases} 0 & \text{dla } g(x, y) < 0 \\ E[(M-1)/2] & \text{dla } g(x, y) = 0 \\ M-1 & \text{dla } g(x, y) > 0 \end{cases}$$

Zastosowanie

obrazy o jednolitym tle i dobrze widocznych obiektach - np. obrazy binarne. Efekt: czarno-biała krawędź na szarym tle.

Metoda **obcinająca**

$$g'(x, y) = \begin{cases} 0 & \text{dla } g(x, y) < 0 \\ g(x, y) & \text{dla } 0 \leq g(x, y) \leq M-1 \\ M-1 & \text{dla } g(x, y) > M-1 \end{cases}$$

76

## Filtry uwypuklające krawędzie

- Zmniejszają udział lub wręcz usuwają informację o obszarach jednorodnych, uwypuklając informację o krawędziach
- Suma współczynników maski równa 0
- Konieczność skalowania

1	0	-1
1	0	-1
1	0	-1

1	-1	-1
1	-2	-1
1	1	1

1	-2	1
-2	4	-2
1	-2	1

77

**Detekcja** (wykrywanie) krawędzi (edge detection)

Jest to technika segmentacji obrazu, polegająca na znajdowaniu pikseli krawędziowych przez sprawdzanie ich sąsiedztwa.

**Krawędź**

Zbiór pikseli na krzywej mający taką właściwość, że piksele

w ich sąsiedztwie, lecz po przeciwnych stronach krzywej mają różne poziomy jasności.

Cel detekcji

znalezienie lokalnych nieciągłości w poziomach jasności obrazu oraz granic obiektów zawartych w obrazie.

78

## Metoda specjalnego gradientu

Stosowana w przypadkach, gdy metody filtracji górnoprzepustowej (FG) powodują wzmocnienie zakłóceń w obszarach leżących wewnątrz konturu.

Zasada

Krawędź uznana jest za istniejącą, jeśli wartość gradientu intensywności w pewnych punktach przekracza ustalony próg. Metody aproksymacji: **Robertsa, Sobela, Prewitta.**

Oznaczenia pikseli:

$f_0$	$f_1$	$f_2$
$f_3$	$f_4$	$f_5$
$f_6$	$f_7$	$f_8$

$(i,j)$

79

## Metoda Robertsa

$$R(i,j) = \sqrt{(f_4 - f_8)^2 + (f_7 - f_5)^2}; \quad \alpha = -\frac{\pi}{4} + \text{tg}^{-1} \left( \frac{f_7 - f_5}{f_4 - f_8} \right)$$

gdzie:

$R(i,j)$  - specjalny gradient w punkcie  $(i,j)$

$\alpha$  - kierunek gradientu intensywności.

Metoda Sobela

dwie składowe gradientu:

$$S_x = (f_2 + 2f_5 + f_8) - (f_0 + 2f_3 + f_6)$$

$$S_y = (f_6 + 2f_7 + f_8) - (f_0 + 2f_1 + f_2)$$

$$S(x,y) = \sqrt{S_x^2 + S_y^2}$$

80

## Maski konwolucyjne

Roberts

1	0	0	-1
0	-1	1	0

$G_x$

$G_y$

Sobel:

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

$G_x$

$G_y$

Prewitt:

1	0	-1	1	1	1
1	0	-1	0	0	0
1	0	-1	-1	-1	-1

$G_x$

$G_y$

81

Przykład:

1	0	0	-1
0	-1	1	0

$G_x$

$G_y$

		$p(i,j)$				$q(i,j)$	
		2	5	6	2	3	
		3	1	3	4	1	
		2	3	5	1	x	
						x	
						x	
						x	

operator  
Robertsa

Maski  
konwolucyjne

$$G_x = p(i, j) - p(i + 1, j + 1);$$

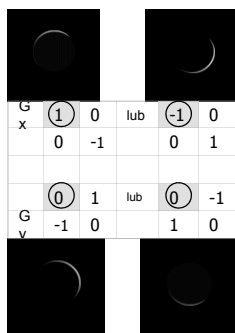
$$G_y = p(i + 1, j) - p(i, j + 1)$$

$$G_{xy} = |G_x| + |G_y|;$$

$$q(1,1) = |2 - 1| + |3 - 5| = 1 + 2 = 3$$

82

## Filtry i operatory Robertsa



$$G = \sqrt{G_x^2 + G_y^2}$$



83

## Uwagi ogólne dotyczące algorytmów

- Dyskusja nad algorytmami obejmuje m.in. zagadnienie złożoności obliczeniowej.
- Podając algorytm próbujemy określić czas i pamięć potrzebne do jego wykonania.
- Typowy błąd: mylenie złożoności obliczeniowej i programowej.
- Generalnie, długość programu realizującego algorytm ma mało wspólnego z szybkością wykonania, a nawet z wymaganą wielkością pamięci.
- Jeśli jakaś relacja istnieje, to jest ona wręcz odwrotna.
- Algorytmy „złożone” są zwykle szybsze niż „proste”. Np. program dla FFT (nierekurencyjny) jest dłuższy i bardziej złożony niż program realizujący wzór sumy dla transformaty. Jednak wykonuje się znacznie szybciej. Podobnych przykładów dostarczają algorytmy sortowania.
- Często atrakcyjniejsze wydaje się użycie rekurencyjnej formy algorytmu, jako dużo krótszej niż nierekurencyjna, a liczba operacji w obu formach jest taka sama. W takich przypadkach należy pamiętać o kosztach wywołań rekurencyjnych, potrzebie przechowywania wartości rejestrów w pamięci itp. Jeśli liczba wywołań jest mała w porównaniu z liczbą innych operacji, to ich koszt może być opłacalny z powodu prostoty programu. W innych przypadkach algorytmy w formie nierekurencyjnej dają programy wydajniejsze.
- O ile prostota programowania może wydawać się atrakcyjna programiście, który jest ograniczony czasem i planuje uruchomienie programu z niewielką ilością danych, o tyle jest **szkodliwa** w przypadkach zastosowań użytkowych przy dużych zbiorach danych.

**Materiał:**

- M.Doros, Przetwarzanie obrazów, skrypt WSISIZ
- Materiały wykładowe POBZ z zeszłego roku na UBIKu
- T.Pavlidis, Grafika i Przetwarzanie Obrazów, WNT Warszawa 1987.
- **I.Pitas**, Digital image processing, algorithms and applications, John Wiley & Sons, Inc. 2000, **pp. 162-166** (w katalogu ...**\APOZ\Materialy** na UBIKu).

**Omówienie tematów projektów**

- **Materiał w katalogach na UBIKu:**  
...**\APOZ\Materialy**  
...**\APOZ\2019-2020\Projekty**