

Przetwarzanie obrazów



Mariusz Borawski
mariusz.borawski@wi.ps.pl

Politechnika Szczecińska
Wydział Informatyki

8 marzec, 2004

Materiały

- 1.Brandt S., Analiza danych, Warszawa 1998;
- 2.Kuchariew G., *Przetwarzanie i analiza obrazów cyfrowych*, Szczecin 1997;
- 3.Pratt W., *Digital image processing*, Wiley Interscience Publication, New York 2001.

Lokalne metody obróbki obrazu

Lokalne metody obróbki zmieniają jasności pikseli na podstawie jasności pikseli sąsiednich. Operacje te mają na ogół na celu polepszenie jakości odbioru obrazu, usunięcie szumów, wydobywanie informacji istotnej itd.

Negatyw

$$P'(x, y) = 255 - P(x, y)$$



Progowanie

Progowanie jest operacją mającą na celu zmniejszenie poziomów jasności obrazka. Możemy je podzielić na:

- binarne;
- binarne odwrotne;
- binarne przedziałowe;
- z zachowaniem poziomów jasności;
- wielopoziomowe.

Progowanie binarne

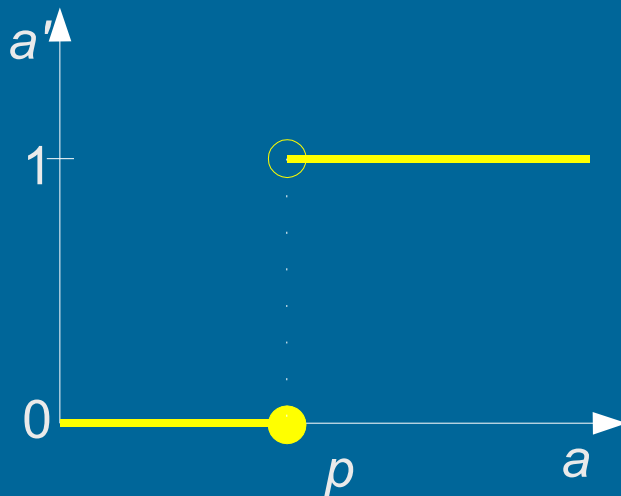
$$a' = \begin{cases} 0 & \text{dla } a \leq p \\ 1 & \text{dla } a > p \end{cases}$$

gdzie

a – jasność piksela obrazka przed progowaniem;

a' – jasność piksela obrazka po progowaniu;

p – próg.



0	4	1	3
5	6	2	6
4	4	1	3
4	1	5	4

$p = 2$

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

Progowanie binarne



Progowanie binarne odwrotne

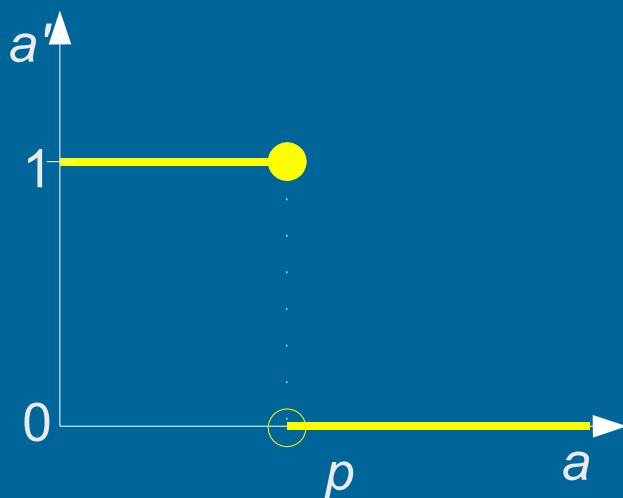
$$a' = \begin{cases} 0 & \text{dla } a > p \\ 1 & \text{dla } a \leq p \end{cases}$$

gdzie

a – jasność piksela obrazka przed progowaniem;

a' – jasność piksela obrazka po progowaniu;

p – próg.



0	4	1	3
5	6	2	6
4	4	1	3
4	1	5	4

$p = 2$
→

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

Progowanie odwrotne binarne



Progowanie binarne przedziałowe

$$a' = \begin{cases} 0 & \text{dla } a < p_1 \vee a > p_2 \\ 1 & \text{dla } p_1 \leq a \leq p_2 \end{cases} \quad \text{lub}$$

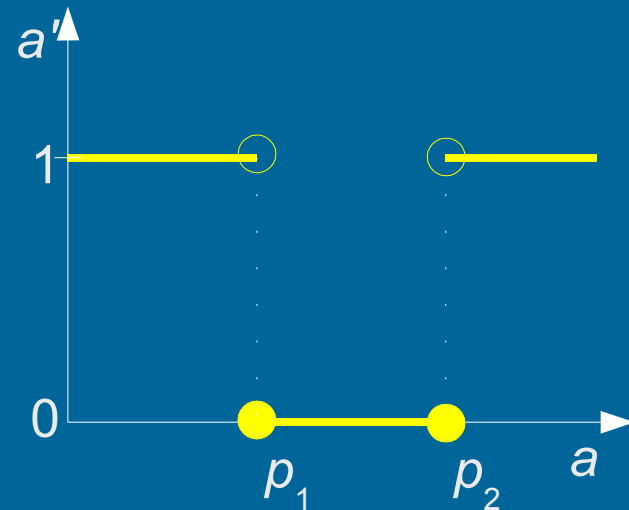
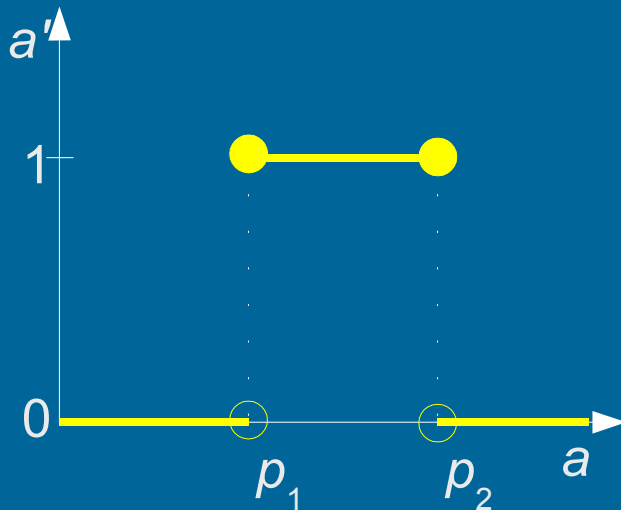
$$a' = \begin{cases} 0 & \text{dla } p_1 \leq a \leq p_2 \\ 1 & \text{dla } a < p_1 \vee a > p_2 \end{cases}$$

gdzie

a – jasność piksela obrazka przed progowaniem;

a' – jasność piksela obrazka po progowaniu;

p_1, p_2 – progi.



Progowanie binarne przedziałowe

0	4	1	3
5	6	2	6
4	4	1	3
4	1	5	4

$p_1 = 2$
 $p_2 = 4$

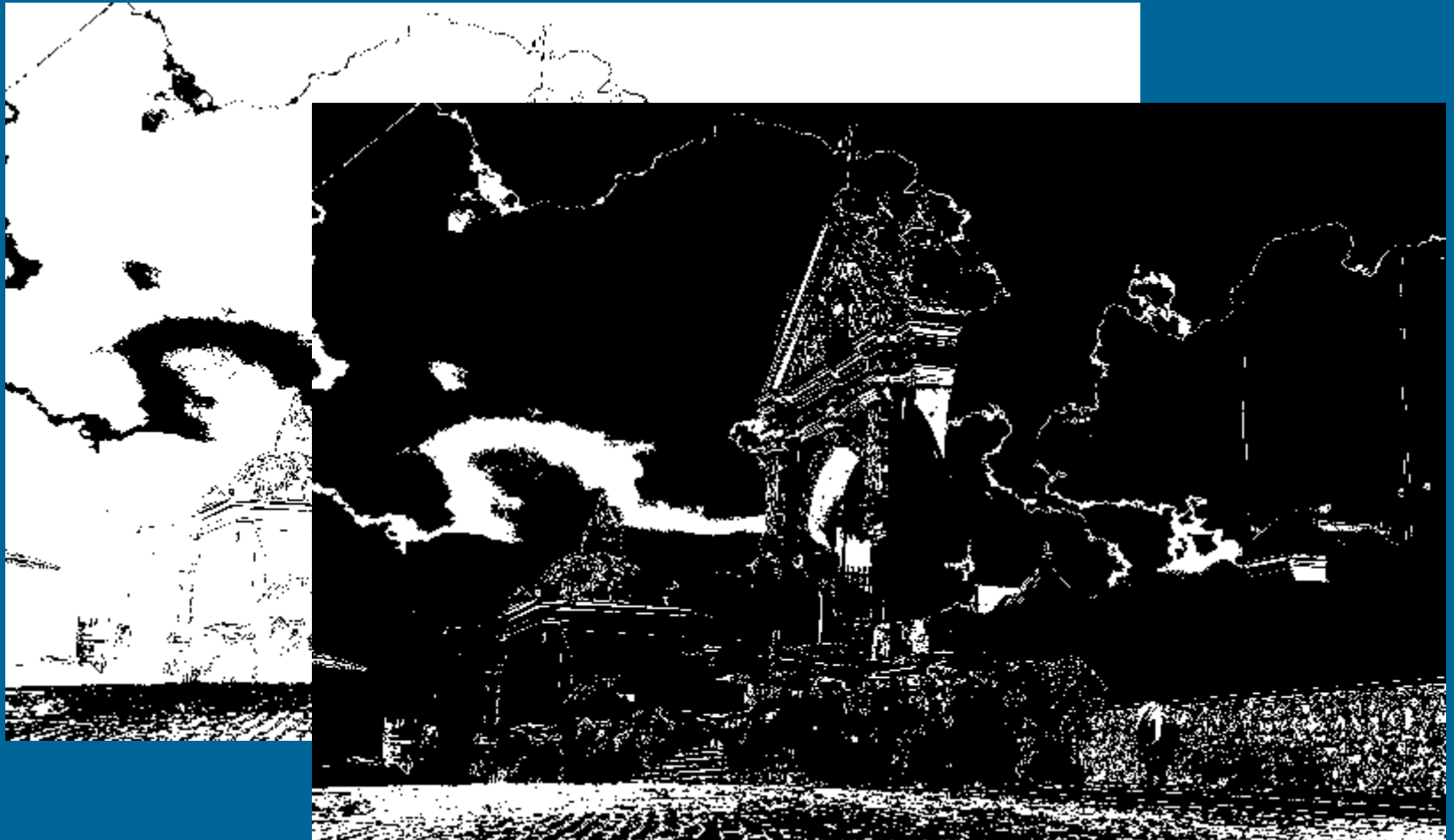
0	1	0	1
0	0	1	0
1	1	0	1
1	0	0	1

0	4	1	3
5	6	2	6
4	4	1	3
4	1	5	4

$p_1 = 2$
 $p_2 = 4$

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

Progowanie białe przedziałowe



Progowanie z zachowaniem poziomów szarości

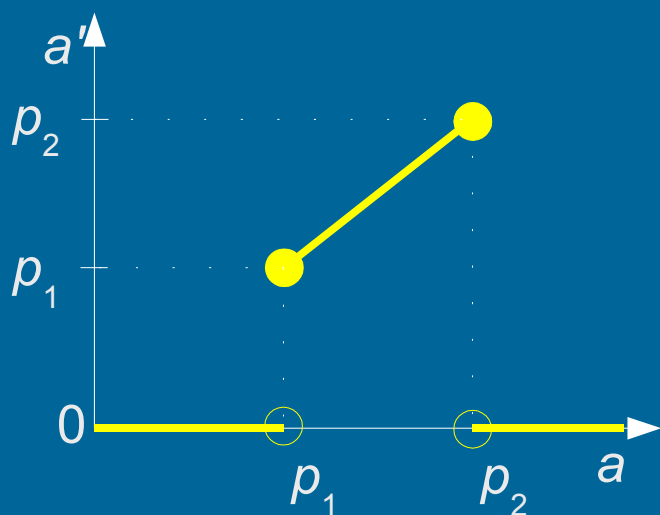
$$a' = \begin{cases} 0 & \text{dla } a < p_1 \vee a > p_2 \\ a & \text{dla } p_1 \leq a \leq p_2 \end{cases}$$

gdzie

a – jasność piksela obrazka przed progowaniem;

a' – jasność piksela obrazka po progowaniu;

p_1, p_2 – progi.



0	4	1	3
5	6	2	6
4	4	1	3
4	1	5	4

$p_1 = 2$
 $p_2 = 4$

0	4	0	3
0	0	2	0
4	4	0	3
4	0	0	4

Progowanie obrazu kolorowego z zachowaniem poziomów jasności

$$A' = \begin{cases} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \text{dla } a_r < p_{r1} \vee a_r > p_{r2} \vee a_g < p_{g1} \vee a_g > p_{g2} \vee a_b < p_{b1} \vee a_b > p_{b2} \\ A & \text{dla } p_{r1} \leq a_r \leq p_{r2} \wedge p_{g1} \leq a_g \leq p_{g2} \wedge p_{b1} \leq a_b \leq p_{b2} \end{cases}$$

gdzie

A – wektor składowych koloru piksela obrazka przed progowaniem;

A' – wektor składowych koloru piksela obrazka po progowaniu;

$p_{r1}, p_{r2}, p_{g1}, p_{g2}, p_{b1}, p_{b2}$ – progi dla poszczególnych kolorów.

Progowanie obrazu kolorowego z zachowaniem poziomów jasności



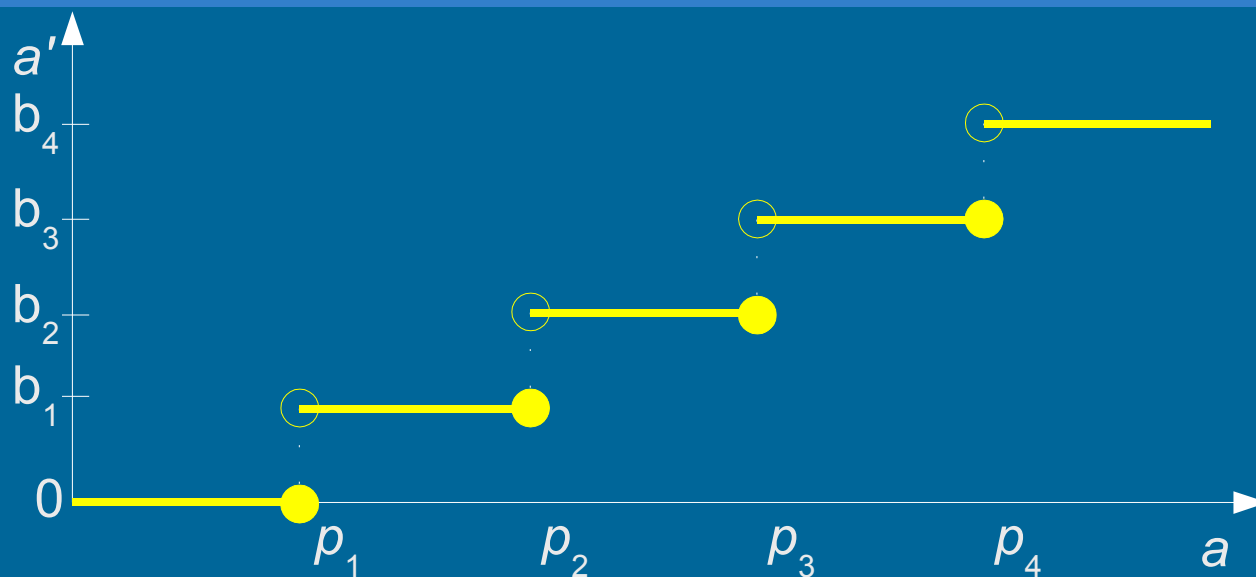
Progowanie wielopoziomowe

$$a' = \begin{cases} 0 & \text{dla } a \leq p_1 \\ b_1 & \text{dla } p_1 < a \leq p_2 \\ b_2 & \text{dla } p_2 < a \leq p_3 \\ \vdots & \\ b_{N-1} & \text{dla } p_{N-1} < a \leq p_N \end{cases}$$

gdzie

- a – jasność piksela obrazka przed progowaniem;
- a' – jasność piksela obrazka po progowaniu;
- b_1, b_2, \dots, b_N – poziomy szarości;
- p_1, p_2, \dots, p_N – progi.

Progowanie wielopoziomowe



0	4	1	3	$p_1 = 2$	0	1	0	1
5	6	2	6	$p_2 = 4$	2	2	0	2
4	4	1	3	\rightarrow	1	1	0	1
4	1	5	4	$b_1 = 1$	1	0	2	1
				$b_2 = 2$				

Progowanie wielopoziomowe



Operacje arytmetyczno-logiczne na obrazach

W pamięci komputera obraz jest dwuwymiarową tablicą wartości liczbowych. Dzięki temu na obrazach można wykonywać operacje matematyczne przewidziane dla macierzy wartości rzeczywistych (dodawanie, odejmowanie) i logicznych.

Dodawanie

Dwa lub więcej obrazów możemy połączyć wykonując prostą operację sumowania macierzy:

$$O' = \frac{O_1}{w_1} + \frac{O_2}{w_2} + \dots + \frac{O_N}{w_N}$$

gdzie

O_1, O_2, \dots, O_N – obrazki sumowane;

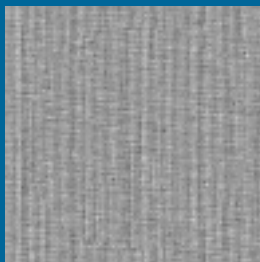
O' – obrazek powstały w wyniku zsumowania;

w_1, w_2, \dots, w_N – współczynniki normalizujące.

Na ogół współczynniki normalizujące muszą spełniać warunek:

$$\frac{1}{w_1} + \frac{1}{w_2} + \dots + \frac{1}{w_N} = 1$$

Dodawanie



Dodawanie

10 Dodawanie\test.m

Odejmnowanie

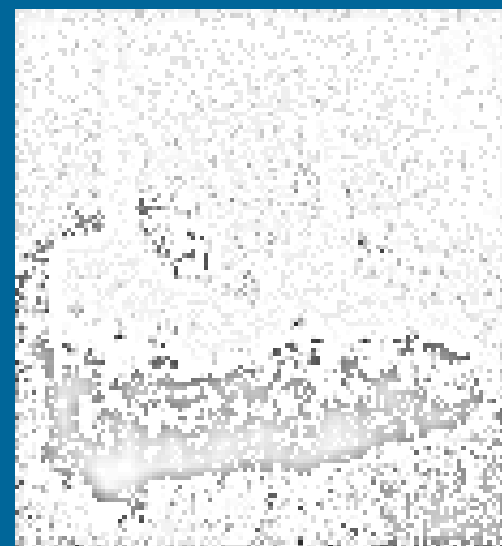
Odejmnowanie obrazów wykorzystuje się w metodzie zwanej nieostrym maskowaniem gdzie od obrazu oryginalnego odejmuje się obraz rozmyty.



-



=



Operacje logiczne

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



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

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

Operacje logiczne

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

xor

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

=

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

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

xor

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

=

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

Operacje logiczne

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

or

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

=

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

Operacje logiczne

08 Konturyzacja\test.m