

# Przetwarzanie obrazów 7

- Handout

SZ 2024/2025

Wydział Matematyki, Fizyki i Informatyki



Dr.-Ing. A. Rowińska-Schwarzsteller  
[agnieszka.rowinska-schwarzsteller@ug.edu.pl](mailto:agnieszka.rowinska-schwarzsteller@ug.edu.pl)

17. Dezember 2024

# Spis treści

<b>13 Wykrywanie krawędzi, grzbietów i kątów w obrazie</b>	<b>2</b>
13.1 Operatory wykorzystywane w detekcji krawędzi . . . . .	6
13.1.1 Operator Laplacian of Gaussian (LoG) . . . . .	10
13.1.2 Operator Difference of Gaussians, DoG . . . . .	14
13.2 Poprawa konturu . . . . .	14
13.2.1 Tłumienie wartości niemaksymalnych ( <i>Non-Maximum Suppression</i> ) . . . . .	17
13.2.2 Progowanie z histerezą ( <i>Hysteresis-Threshold</i> ) . . . . .	20
13.3 Detektor krawędzi Canny'ego ( <i>Canny edge detector</i> ) . . . . .	21
13.4 Detekcja krawędzi w przestrzeni RGB . . . . .	22
13.5 Detekcja grzbietów ( <i>Ridge Detection</i> ) . . . . .	28
13.5.1 Wykrywania grzbietów z macierzą Hessego . . . . .	29
13.6 Detekcja kątów obiektów ( <i>Corner detection</i> ) . . . . .	35
13.6.1 Harris Corner Detectior . . . . .	35
<b>14 Segmentacja obiektów</b>	<b>40</b>
14.0.1 Segmentacja: rozkład obrazu na jednostki semantyczne . . . . .	40
14.1 Metody punktowe . . . . .	43
14.1.1 Globalne metody progowania . . . . .	43
14.1.2 Progowanie lokalne . . . . .	46
14.2 Metody krawędziowe . . . . .	47
14.2.1 Transformacja wododziałowa ( <i>Watershed Transformation</i> , WTH) . . . . .	48
14.3 Metody obszarowe . . . . .	53
14.3.1 Strategie wieloskalowe . . . . .	53
14.3.2 Rozrost obszarów ( <i>Region Growing</i> ) . . . . .	56
14.3.3 Łączenie obszarów ( <i>Region Merging</i> ) . . . . .	61
14.3.4 Podział i łączenie ( <i>Split &amp; Merge</i> ) . . . . .	63
14.3.5 Klastrowanie . . . . .	65
14.4 Tekstura . . . . .	69
14.4.1 Macierz szarości i miary tekstuury Haralicka ( <i>Co-Occurrence-Matrix</i> , <i>Haralick features</i> ) . . . . .	72

# 13

## Wykrywanie krawędzi, grzbietów i kątów w obrazie

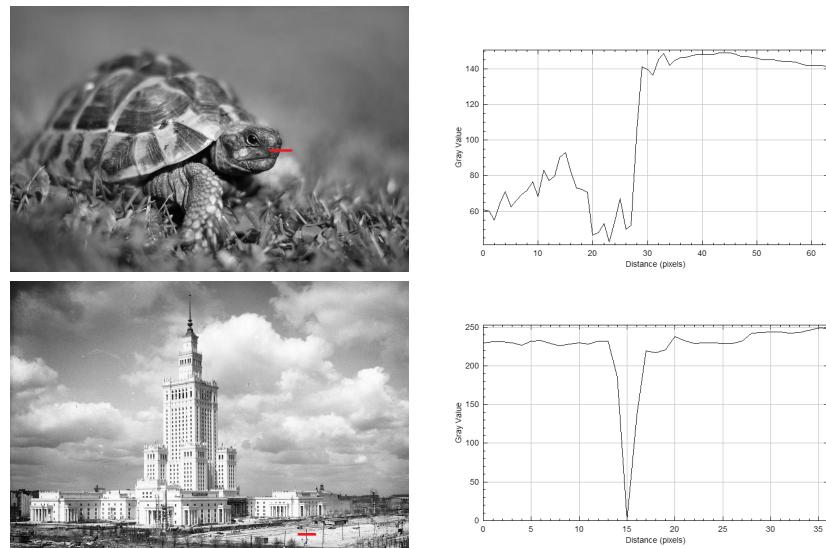
Wykrywanie krawędzi jest podstawowym narzędziem w przetwarzaniu obrazu, widżeniu maszynowym i widzeniu komputerowym, szczególnie w obszarach wykrywania i ekstrakcji cech.

- Krawędź to granica pomiędzy dwiema jednorodnymi powierzchniami obraz —> grafika 13.1

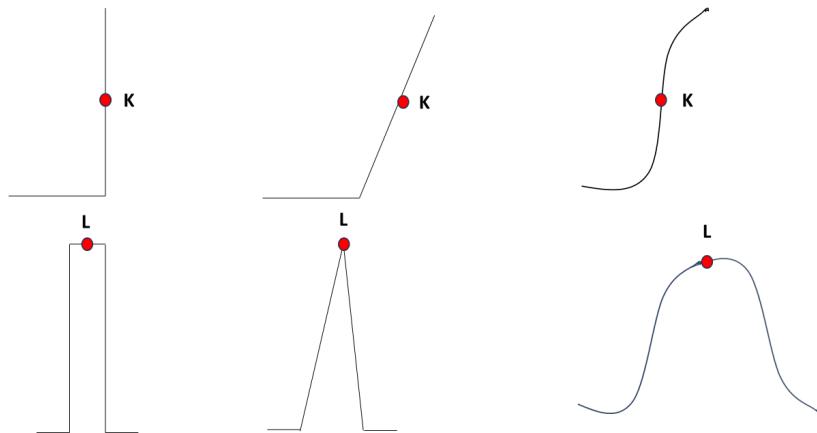


Grafika 13.1: Krawędzie na obrazie.

- Rozróżnia się krawędzie wartości szarości (intensywności, luminancji), krawędzie tekstuury i koloru.
- Krawędzie dla wartości szarości:
  - > nieciągłość wartości szarości (grafika 13.2)
  - na podstawie zmian wartości szarości można zdefiniować różne modele profilu krawędzi. —> grafika 13.3



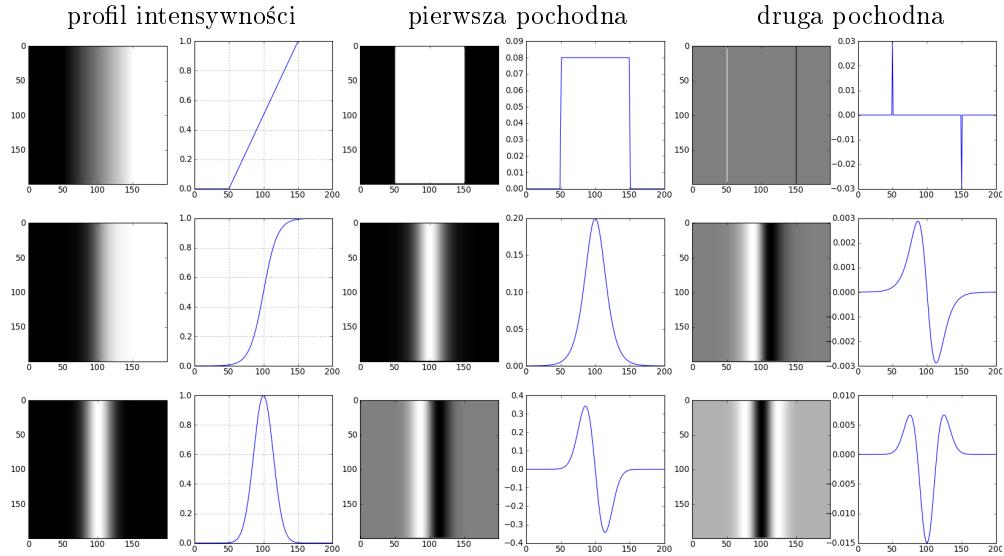
Grafika 13.2: Rzeczywiste krawędzie i linie na obrazie.



Grafika 13.3: Modelowanie krawędzi i linii. Idealny detektor powinien wykryć oznaczony punkt na profilu.

## Krawędzie a szum

Obrazy idealnych profili intensywności linii i krawędzi i ich pochodne → grafika 13.4



Grafika 13.4: Idealne profile intensywności i ich pochodne. Źródło: M. Rohs, MHCI Lab

Szum Gaussa  $\sigma = 0.0001$  w profilu intensywności linii i krawędzi → grafika 13.5

Szum Gaussa  $\sigma = 0.001$  w profilu intensywności linii i krawędzi → grafika 13.6

Szum Gaussa  $\sigma = 0.01$  w profilu intensywności linii i krawędzi → grafika 13.7

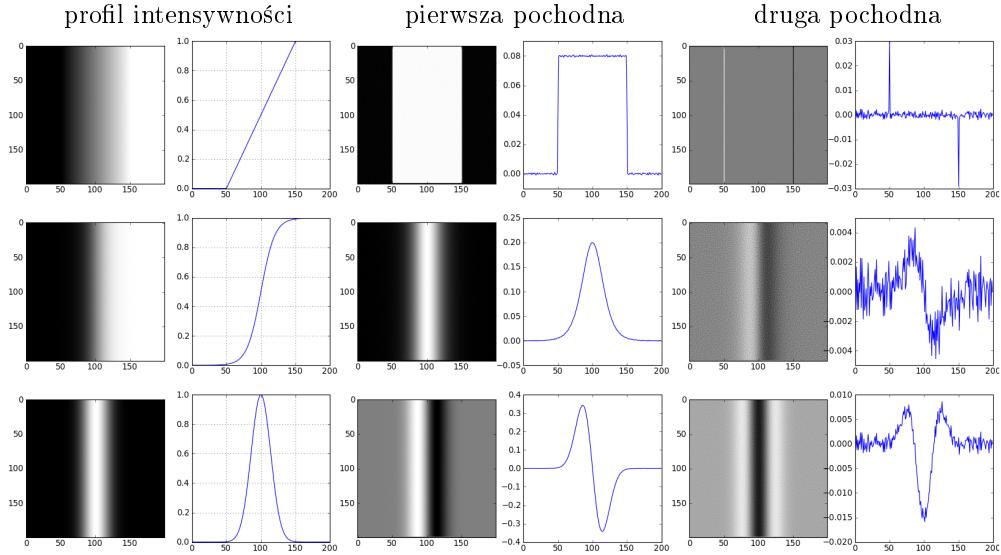
## Orientacja krawędzi

- Krawędzie mogą być zorientowane w dowolny sposób.
- Tworzenie różnicy (pochodnej) w dwóch ortogonalnych kierunkach. → grafika 13.8

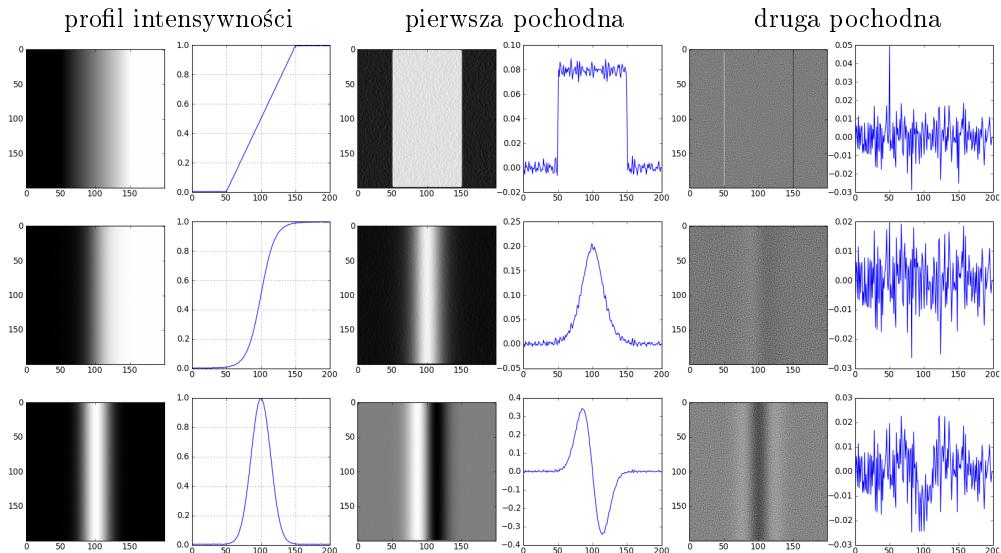
## Ogólna procedura ekstrakcji krawędzi:

Trudność w wykrywaniu krawędzi polega na rozróżnieniu przypadkowych zmian sygnału spowodowanych szumem od poszukiwanych nieciągłości wartości, które odpowiadają rzeczywistym krawędziom.

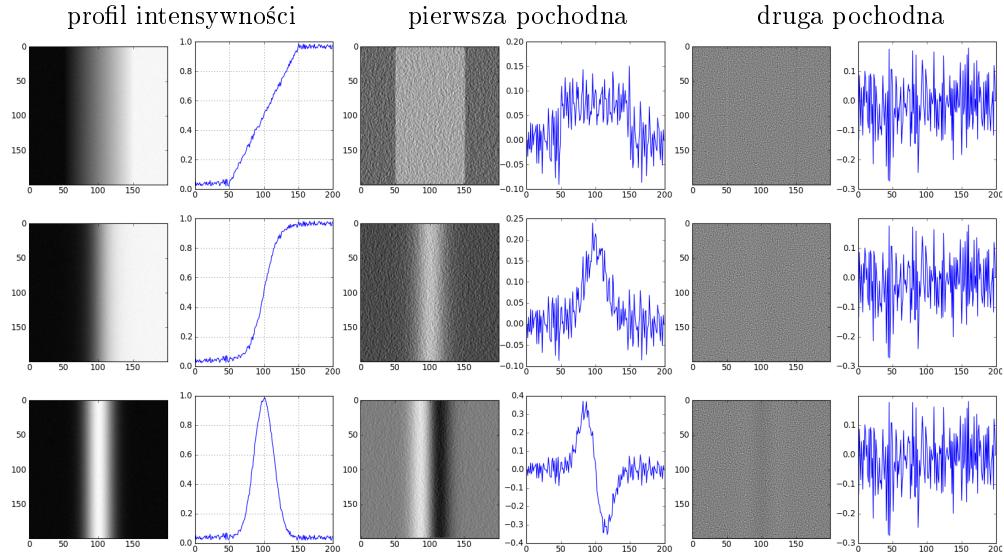
1. wygładzanie (filtr dolnoprzepustowy)  
↪ ponieważ oryginalny obraz jest zwykle zaszumiony, bez wygładzenia wykryte byłoby zbyt wiele artefaktów



Grafika 13.5: Szum Gaussa  $\sigma = 0.0001$  w profilu intensywności linii i krawędzi. Źródło: M. Rohs, MHCI Lab



Grafika 13.6: Szum Gaussa  $\sigma = 0.001$  w profilu intensywności linii i krawędzi. Źródło: M. Rohs, MHCI Lab



Grafika 13.7: Szum Gaussa  $\sigma = 0.01$  w profilu intensywności linii i krawędzi. Źródło: M. Rohs, MHCI Lab

2. wykrycie krawędzi  
→ zastosowanie algorytmu wykrywania krawędzi
  3. poprawa konturu  
→ wykryte punkty krawędzi są rozrzedzane, krótkie krawędzie są eliminowane
  4. powiązanie punktów konturu  
→ łączenie punktów konturu w celu utworzenia zamkniętych linii krawędzi
- grafika 13.9

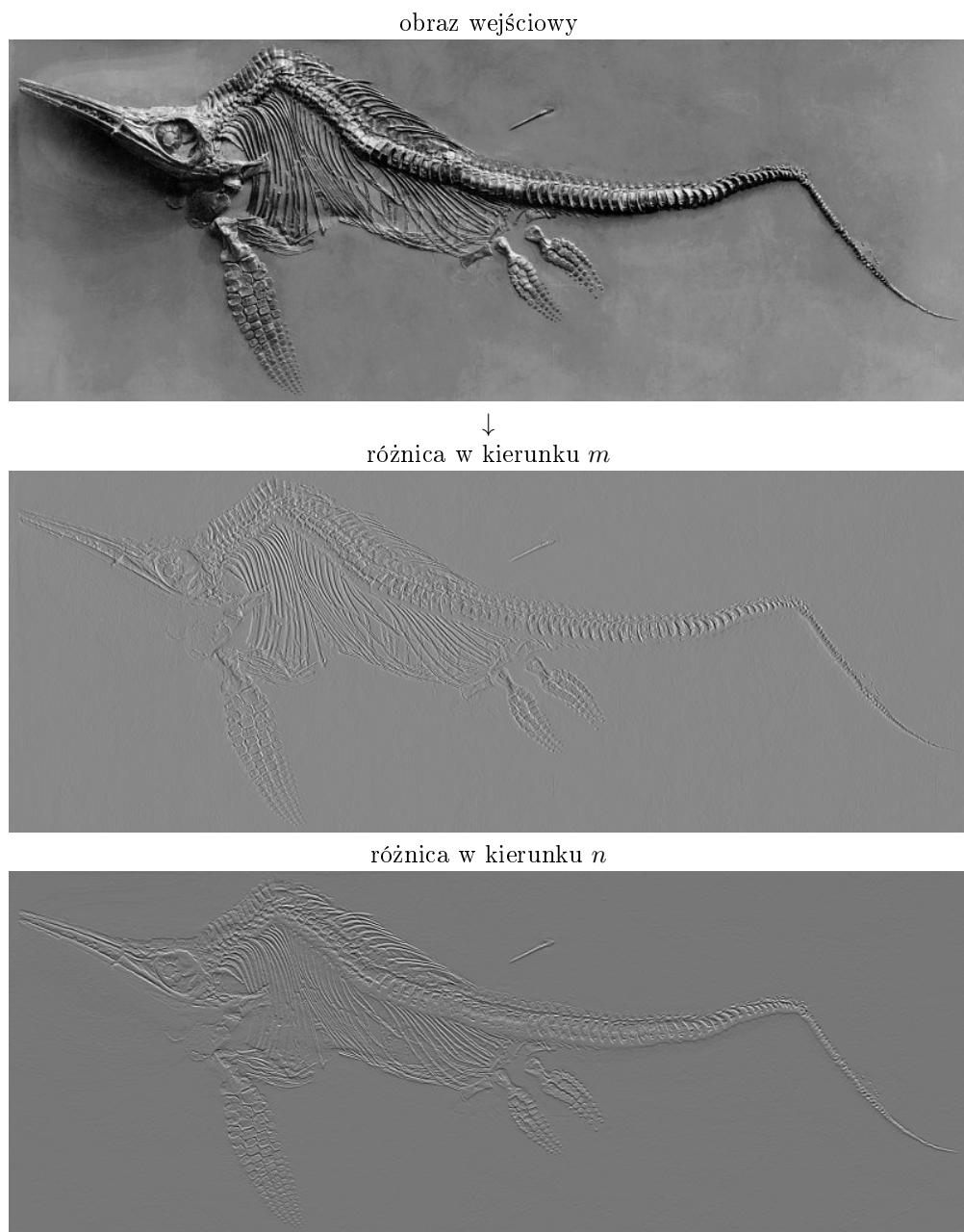
### 13.1 Operatory wykorzystywane w detekcji krawędzi

→ filtry liniowe

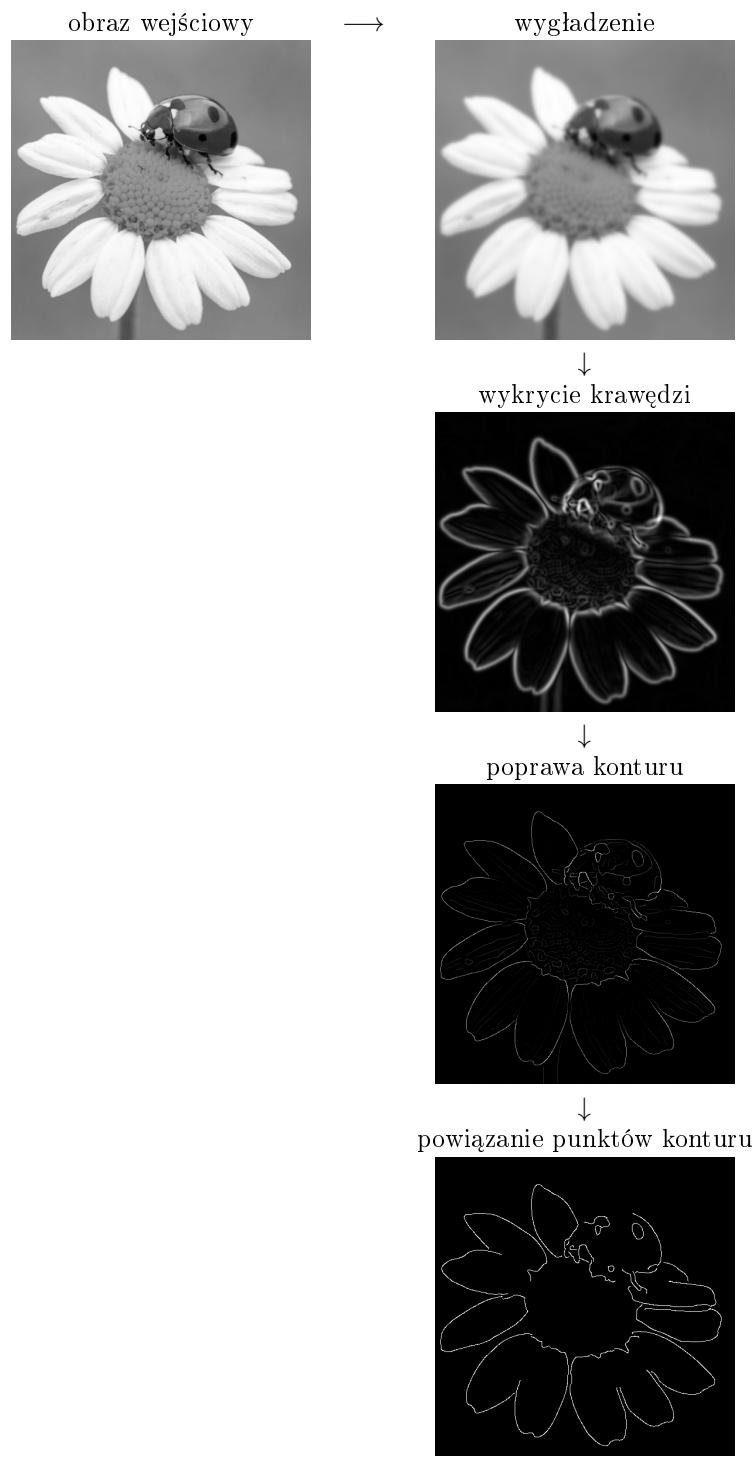
Operatory do obliczania pierwszej pochodnej:

- Sobel
- Prewitt
- Roberts
- Kirsch

**Przykład:** znajdywanie krawędzi filtrem Sobel'a w ImageJ ("Find Edges")  
→ grafika 13.10



Grafika 13.8: Orientacja krawędzi: tworzenie różnicy w kierunku  $m$  i w kierunku  $n$



Grafika 13.9: Algorytmy ekstrakcji krawędzi

Dwa jądra splotu  $h_1$  i  $h_2$  generują pochodne pionowe i poziome w obrazie wejściowym  $g$ :

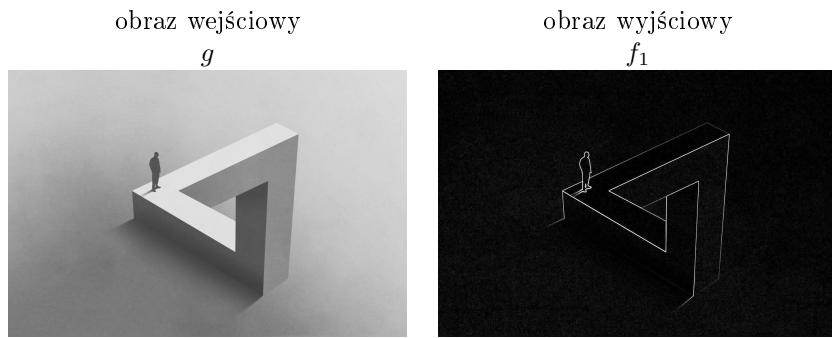
$$h_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad h_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Obraz wyjściowy  $f_{1,2}$  obliczany jest ze wzoru

$$f_1 = \sqrt{(g * h_x)^2 + (g * h_y)^2}$$

lub

$$f_2 = |g * h_x| + |g * h_y|$$



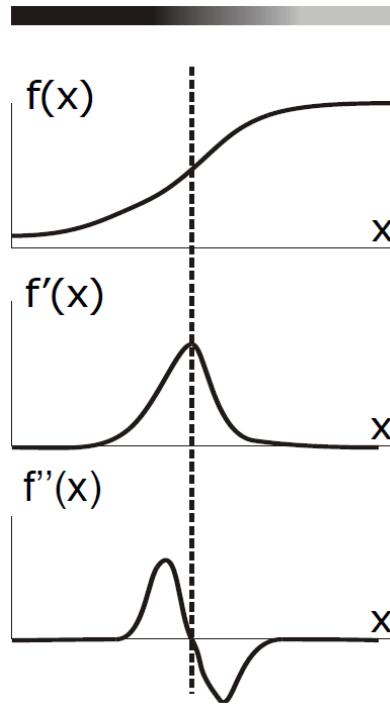
Grafika 13.10: Wykrywanie krawędzi filtrem Sobel'a

Operatory do obliczania drugiej pochodnej:

- Laplace
- LoG
- DoG

### Zalety operatorów operujących na drugiej pochodnej

- łatwiej rozpoznać zmianę znaku (przejście przez zero) niż minimum lub maksimum funkcji:



- wykrywanie krawędzi można przeprowadzić na dwa różne sposoby:
  - w filtrowanym obrazie wynikowym można szukać przejść przez zero (*Zero-crossings*), czyli miejsc zmiany znaku,
  - do wyników filtrowania można zastosować metodę wartości progowej (ta metoda ze względu na swoją prostotę jest stosowana w praktyce)

**Przykład:** wykrywanie krawędzi filtrem Laplace'a

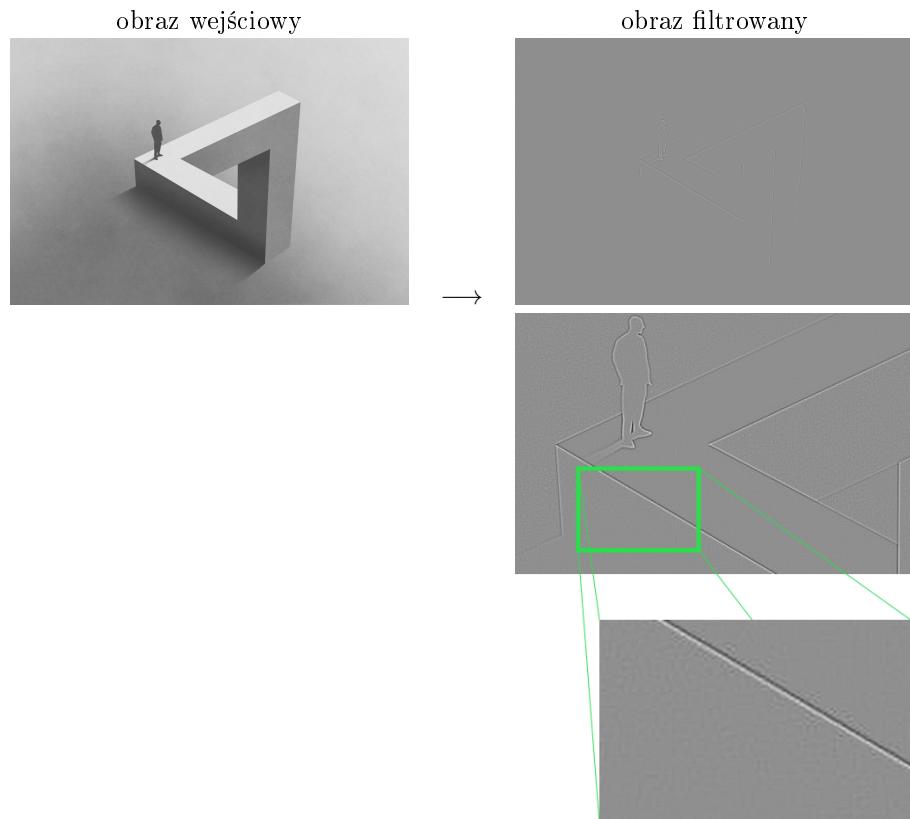
$$h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Wyznaczenie przejść pochodnej przez 0 → grafika 13.11

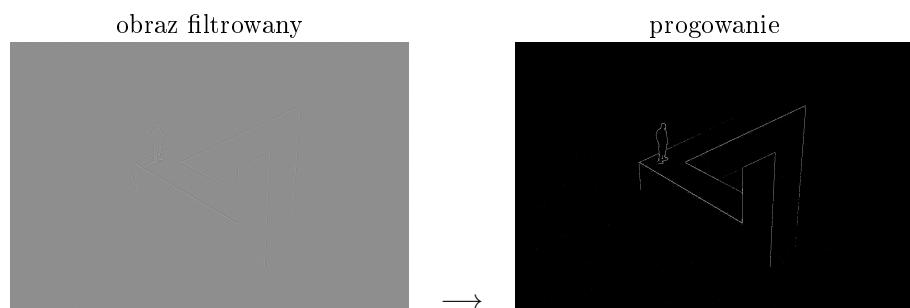
Progowanie obrazu wynikowego → grafika 13.12

#### 13.1.1 Operator Laplacian of Gaussian (LoG)

- połączenie filtra dolnoprzepustowego (wygładzanie) z filtrem górnoprzepustowym (wzmacnianie kontrastu)
- jądro filtra jest tworzone poprzez zastosowanie operatora Laplace'a do funkcji Gaussa
- w literaturze także nazywany Marr-Hildreth-Operator lub Mexican Hat



Grafika 13.11: Przejścia przez zero operatora Laplace'a: szarość o średniej wartości = 0, ciemne piksele < 0, jasne piksele > 0.



Grafika 13.12: Progowanie obrazu filtrowanego filtrem Laplace'a.

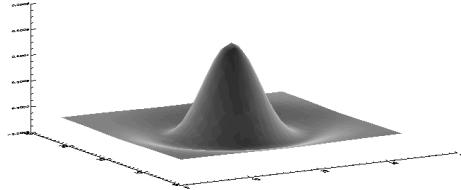
### Funkcja Laplace'a funkcji Gaussa (LoG)

Funkcja Gaussa 2D:

$$H_G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Operator Laplace'a (operator różniczkowy drugiego rzędu)  $\nabla^2$ :

$$\begin{aligned} h_{LoG}(x, y) &= \nabla^2 H_G(x, y) = \frac{\partial^2 H_G(x, y)}{\partial x^2} + \frac{\partial^2 H_G(x, y)}{\partial y^2} \\ &= -\frac{1}{\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) \end{aligned}$$



Aby móc korzystać z tej funkcji w przetwarzaniu obrazu, ciągły LoG jest aproksymowany dyskretnie. Aproksymację przeprowadza się dla jąder filtrów o nieparzystych długościach krawędzi  $k = 3, 5, 7, \dots$

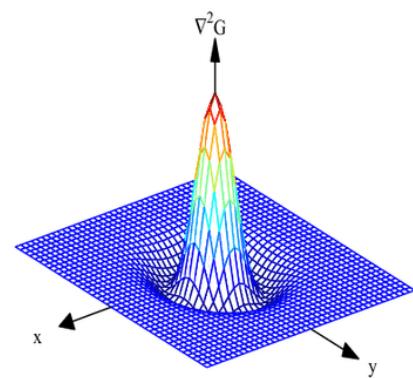
Przykładowe jądro o wielkości  $7 \times 7$ , tj. dyskretne przybliżenie ciągłego LoG z odchyleniem standardowym  $\sigma = 1.6$  (jądro znormalizowane do 1, a następnie pomnożone przez 255):

$$h(m, n) = \begin{pmatrix} 1 & 3 & 4 & 4 & 4 & 3 & 1 \\ 3 & 4 & 3 & 0 & 3 & 4 & 3 \\ 4 & 3 & -9 & -17 & -9 & 3 & 4 \\ 4 & 0 & -17 & -30 & -17 & 0 & 4 \\ 4 & 3 & -9 & -17 & -9 & 3 & 4 \\ 3 & 4 & 3 & 0 & 3 & 4 & 3 \\ 1 & 3 & 4 & 4 & 4 & 3 & 1 \end{pmatrix}$$

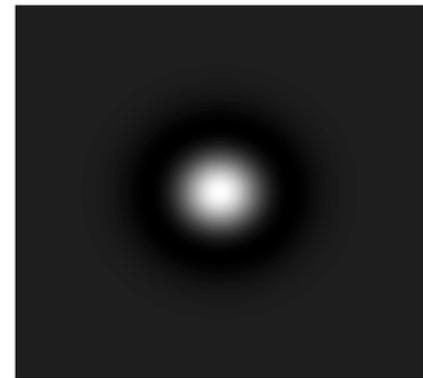
Przykład: → grafika 13.1.3

### Zalety operatora LoG

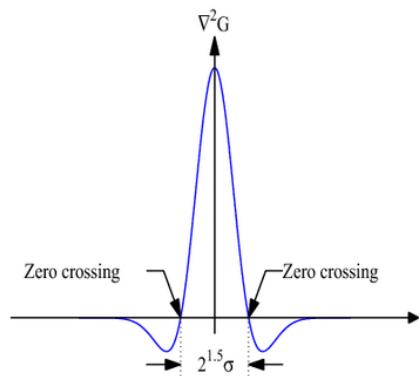
- wygładzanie i podkreślanie krawędzi w jednym operatorze
- mniejsza wrażliwość na szum dzięki wygładzaniu
- funkcja Gaussa nie generuje efektu Gibbsa
- niezmiennik rotacji
- separowalny



**(a)** 3-D plot of the negative of the LoG.



**(b)** Negative of the LoG displayed as an image.



**(c)** Cross section of (a) showing zero-crossing.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

**(d)**  $5 \times 5$  mask approximation to the shape in (a). The negative of this mask would be used in practice.

Grafika 13.13: Operator LoG  $5 \times 5$ ,  $\sigma = 1.5$ . Źródło: Pnachi Li, "Quantum implementation of classical Marr–Hildreth edge detection", Quantum Information Processing 19(2)

### Algorytm wykrywania krawędzi Marr-Hildreth

$$g'(x, y) = [\nabla^2 H_G(x, y)] * g(x, y) = \nabla^2 [H_G(x, y) * g(x, y)]$$

1. filtr Gaussa  $n \times n$

2. filtr Laplace'a  $3 \times 3$

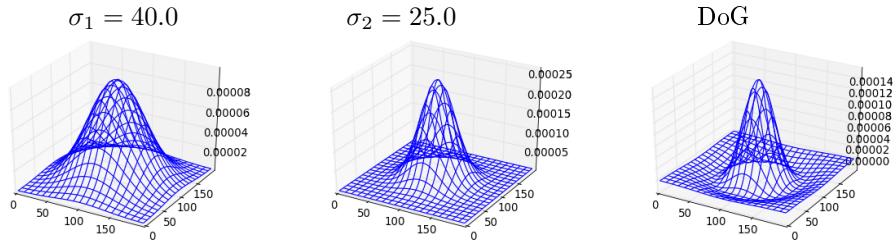
$$h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

3. znalezienie przejść przez zero (*Zero-crossings*) lub progowanie

Przykład: → grafika 13.14

#### 13.1.2 Operator Difference of Gaussians, DoG

- aproksymacja LoG
- różnica dwóch filtrów Gaussa o różnej wariancji



$$h_{DoG}(x, y) = \frac{1}{\pi\sigma_1^4} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{\pi\sigma_2^4} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

Przykład: → grafika 13.15

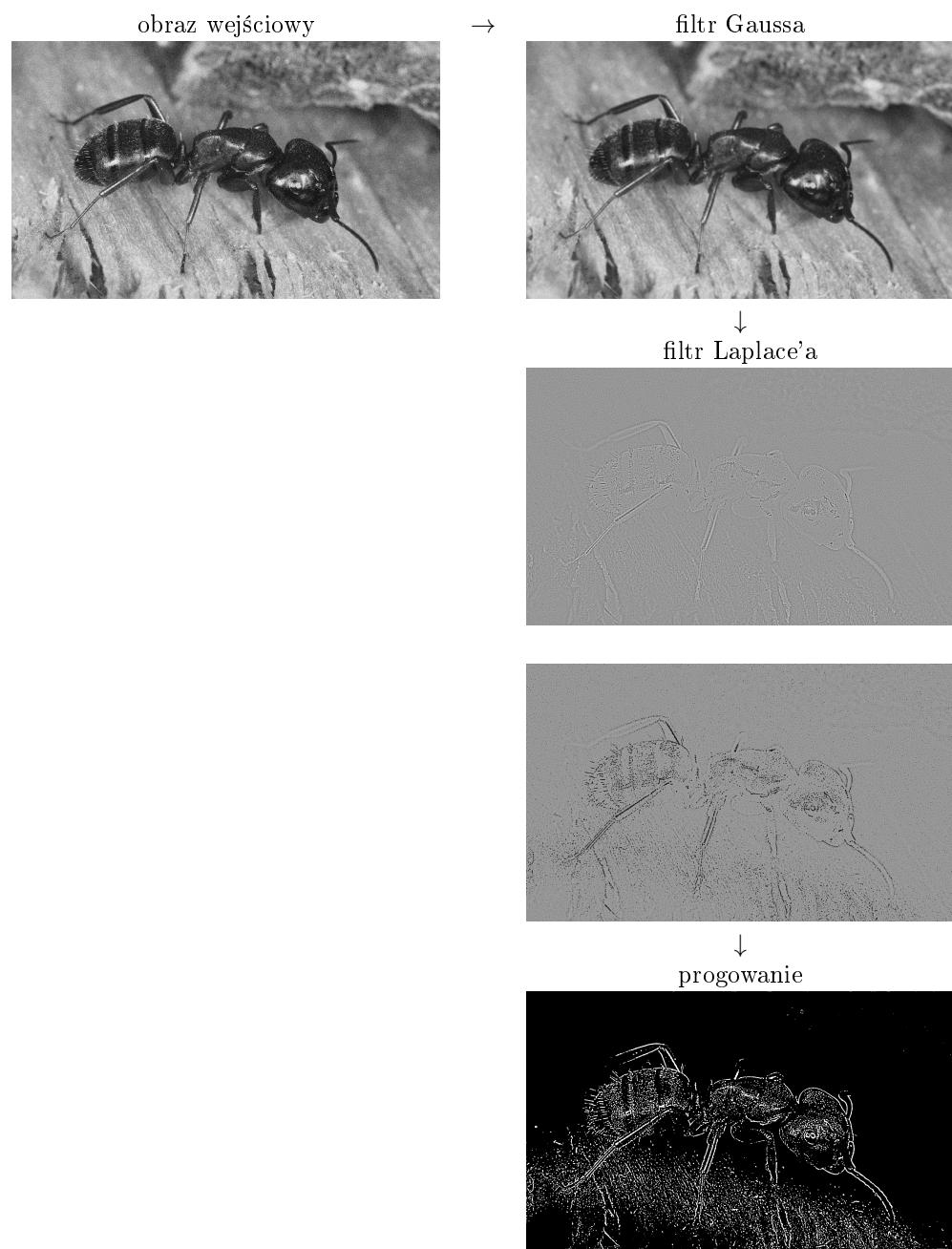
## 13.2 Poprawa konturu

Operatory różnicowe (krawędziowe) dostarczają obraz gradientowy z informacjami o potencjalnych punktach krawędziowych, takimi jak: wartość gradientu i kierunek krawędzi (gradientu).

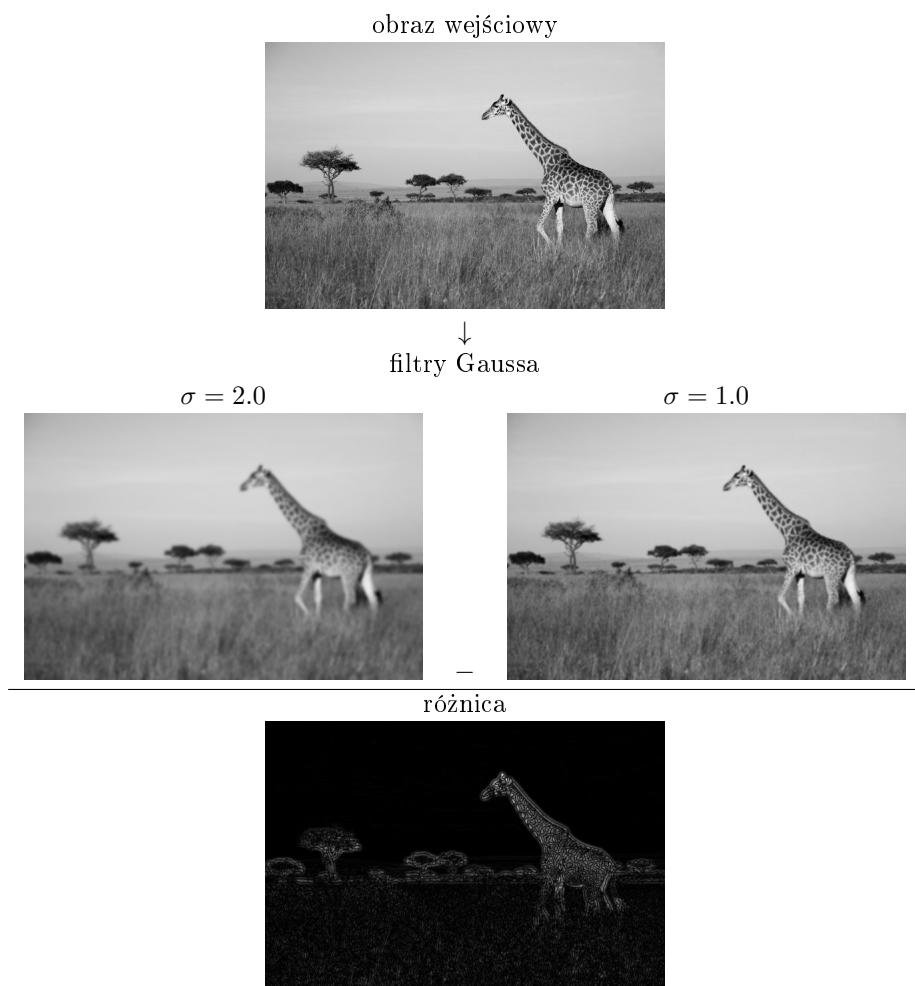
### Gradient

Zastosowanie filtra różnicowego (np. Sobel'a) wyznacza różnicę w dwóch kierunkach ortogonalnych dla każdego piksela.

$$\begin{aligned} g_x(m, n) &= g(m, n) * h_x(m, n) \\ g_y(m, n) &= g(m, n) * h_y(m, n) \end{aligned}$$



Grafika 13.14: Algorytm wykrywania krawędzi Marr-Hildreth dla zaszumionego obrazu. Wykryta dodatnia i ujemna pochodna podkreślone kolorami.



Grafika 13.15: Wykrywanie krawędzi operatorem DoG

Każdemu pikselowi  $g(m, n)$  przypisane są dwie wartości gradientu, które są łączone w celu wyznaczenia wielkości gradientu.

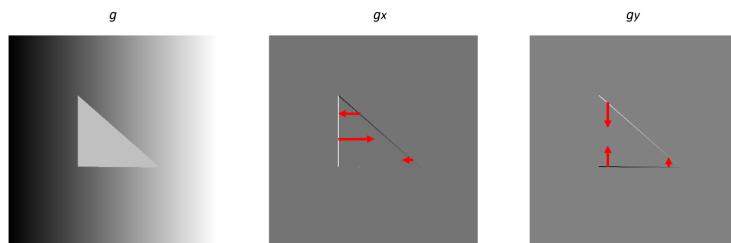
Wartość gradientu  $G$  oblicza się jako:

$$G(m, n) = \sqrt{g_x^2(m, n) + g_y^2(m, n)}$$

$$\approx |g_x(m, n)| + |g_y(m, n)|$$

Kierunek gradientu  $R$  oblicza się ze wzoru:

$$\theta(m, n) = \arctan \frac{g_y}{g_x}$$

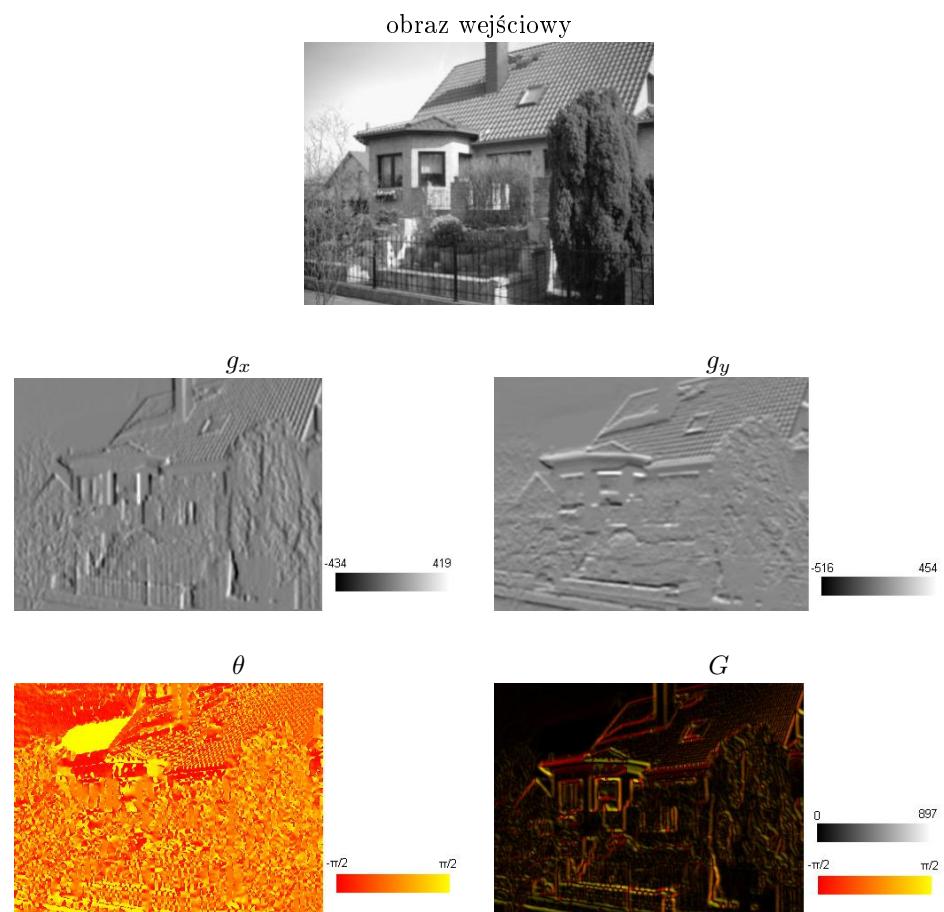


**Przykład:** gradient wyznaczony przez operatory Sobel'a → grafika 13.16

### 13.2.1 Tłumienie wartości niemaksymalnych (*Non-Maximum Suppression*)

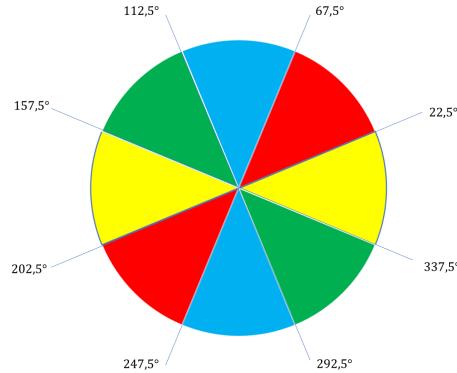
- W wielu metodach wykrywania krawędzi stosuje się stały próg, aby zdecydować, czy w tym miejscu znajduje się krawędź, czy nie. W zależności od kształtu krawędzi tworzy to większe obszary, na których może znajdować się ewentualna krawędź.
- W przypadku tłumienia wartości niemaksymalnych gradientu jego wartość porównywana jest poprzecznie do kierunku krawędzi.
  - ↪ Punkt konturu o maksymalnej wartości pozostaje niezmieniony, natomiast wszystkie pozostałe ustawiane są na wartość 0, czyli są pomijane w dalszym przetwarzaniu (stąd nazwa tłumienie "nonmaxima").
- Istnieją tylko cztery możliwe kierunki opisu otaczających pikseli -  $0^\circ$  (w kierunku poziomym),  $45^\circ$  (wzdłuż dodatniej przekątnej),  $90^\circ$  (w kierunku pionowym) lub  $135^\circ$  (wzdłuż ujemnej przekątnej).

x	x	x	x	x
x	g	b	r	x
x	y		y	x
x	r	b	g	x
x	x	x	x	x

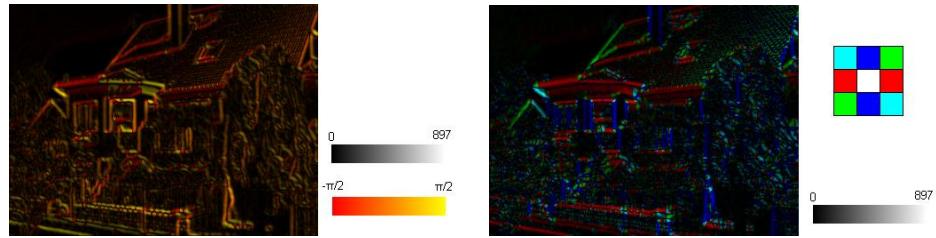


Grafika 13.16: Gradient wyznaczony przez operatory Sobel'a.  
Źródło: commons.wikimedia.org

- Możliwe przedziały dla wartości kierunku gradientu  $\theta$ :



**Przykład:** gradient wyznaczony przez operatory Sobel'a i przypodziałkowanie możliwych kierunków krawędzi → grafika 13.17



Grafika 13.17: Gradient wyznaczony przez operatory Sobel'a i przypodziałkowanie możliwych kierunków krawędzi. Źródło: commons.wikimedia.org

### Algorytm Non-Maximum Suppression

1. Wybór dwóch sąsiadów (sąsiedztwo 8-spójne) aktualnego piksela, tak aby spełniony był warunek "*w poprzek do kierunku krawędzi*".

Temu celowi służą poniższe maski. Wybór maski zależy od kierunku gradientu  $\theta$  bieżącego piksela:

$$(a) \quad \theta \in [337.5^\circ, 22.5^\circ) \quad \theta \in [22.5^\circ, 67.5^\circ) \quad \theta \in [67.5^\circ, 112.5^\circ) \quad \theta \in [112.5^\circ, 157.5^\circ) \\ \cup [157.5^\circ, 202.5^\circ) \quad \cup [202.5^\circ, 247.5^\circ) \quad \cup [247.5^\circ, 292.5^\circ) \quad \cup [292.5^\circ, 337.5^\circ)$$

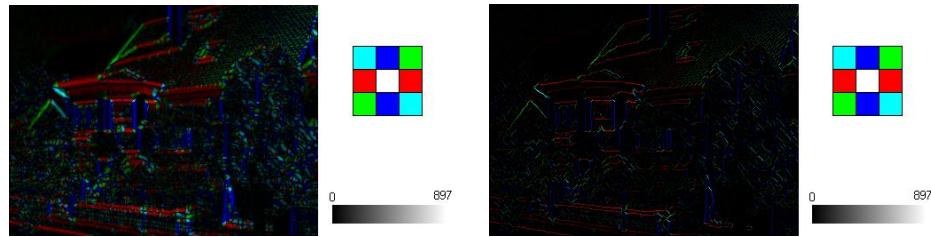


2. Jeżeli odchylenia kierunków gradientu aktualnego piksela i jego dwóch sąsiadów przekraczają daną wartość progową:

→ przejście do następnego aktualnego piksela i rozpoczęcie od pkt. 1.

3. Jeżeli wartość gradientu aktualnego piksela jest nie większa niż jego sąsiadów, wartość gradientu zostaje stłumiona ( $=: 0$ ),  
→ przejście do następnego aktualnego piksela i rozpoczęcie od pkt. 1.
4. Optionalnie: jeśli wartość gradientu aktualnego piksela jest poniżej (niskiego) progu, zostaje on wytłumiony nawet przy lokalnym maksymum.

**Przykład:** tłumienie wartości niemaksymalnych gradientu → grafika 13.18



Grafika 13.18: Tłumienie wartości niemaksymalnych gradientu (lokalne maksyma).  
Źródło: commons.wikimedia.org

### 13.2.2 Progowanie z histerezą (Hysteresis-Threshold)

Poprzez progowanie wartości gradientu dla zbioru potencjalnych punktów krawędziowych podejmowana jest decyzja, czy punkt jest punktem krawędziowym.

Metoda Hysteresis-Threshold nie wykorzystuje pojedynczej wartości progowej, ale przedział wartości progowych ( $T_1, T_2$ ),  $T_2 > T_1$ .

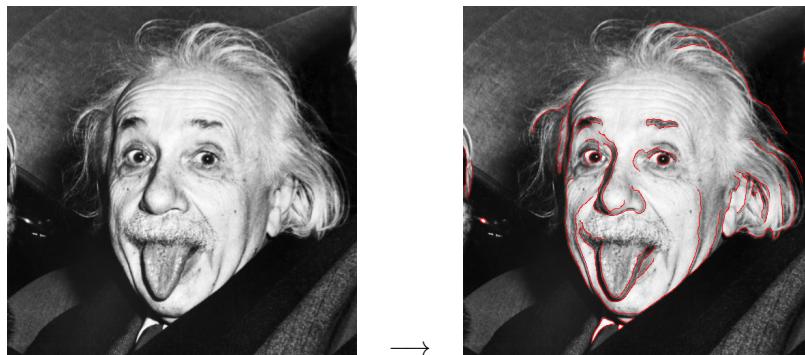
- Punkty, w których wartość gradientu przekracza próg  $T_2$ , służą do rozpoczętania nowych krawędzi.
- Punkty, w których wartość gradientu należy do przedziału ( $T_1, T_2$ ), służą do kontynuacji krawędzi.
- Punkty, w których wartość gradientu jest poniżej progu  $T_1$  są odrzucone.
- Wybór progu (często stosowany w praktyce):
  - $T_2$  wyznaczony metodą Otsu
  - $T_1 = \frac{1}{2}T_2$

Dwie obserwacje są interesujące w ocenie skuteczności metody Hysteresis-Threshold:

- kontur, na którym wszystkie wartości gradientu należą do przedziału ( $T_1, T_2$ ) , nie jest rozpoznawany,
- pojedynczy punkt o wartości gradientu  $> T_2$  wystarczy, aby utworzyć kontur, nawet gdy wszystkie inne punkty tego konturu mają wartości gradientu  $< T_2$ .

### 13.3 Detektor krawędzi Canny'ego (*Canny edge detector*)

→ grafika 13.19



Grafika 13.19: Detektor krawędzi Canny'ego.

Algorytm Canny'ego to wielostopniowy algorytm wykrywania krawędzi, który jest szeroko stosowany w cyfrowym przetwarzaniu obrazu.

Główne kryteria detekcji krawędzi dla algorytmu są następujące:

- dobra detekcja (niski poziom błędów)
  - krawędzie pojawiające się na obrazie nie są pomijane i nie wykrywa się fałszywych krawędzi
- dobra lokalizacja krawędzi
  - odległość pomiędzy pikselami znalezionymi przez detektor a rzeczywistą krawędzią musi być minimalna
- pojedyncza odpowiedź
  - każda krawędź jest rozpoznawana tylko raz.

#### Kroki algorytmu

→ grafika 13.20

##### 1. Redukcja szumu

- filtr Gaussa
- filtr o współczynnikach dwumianu Newtona

##### 2. Wyznaczenie gradientu obrazu

- operator Robertsa
- operator Prewitta
- operator Sobel'a

→ przyporządkowanie możliwych kierunków krawędzi ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ )

3. Usunięcie niemaksymalnych pikseli (Non-Maximum Suppression)
4. Progowanie z histerezą (Hysteresis-Threshold)

**Problemy klasycznych metod wykrywania krawędzi:**

1. wykrywanie cienkich linii (struktur) w obrazie - "grzbiety i doliny" (dwie lokalnie równoległe krawędzie) → grafika 13.21
2. lokalizacja kątów obiektów

### 13.4 Detekcja krawędzi w przestrzeni RGB

Detekcja krawędzi w obrazach kolorowych zwykle odbywa się w poszczególnych kanałach kolorów niezależnie od siebie, a wyniki są łączone za pomocą operatora punktowego.

Możliwe operatory punktowe wyznaczające odległości kolorów to

- norma  $L_1$  (suma wartości),
- norma  $L_2$  (odległość euklidesowa),
- norma  $L_\infty$  (maksimum wartości).

Przy zastosowaniu algorytmu Canny'ego połączenie kanałów RGB może być zastosowane przed każdym z kolejnych kroków algorytmu:

1. wyznaczenie gradientów kierunkowych obrazu
2. wyznaczenie wartości i kierunku gradientu
3. Non-Maximum Suppression (usunięcie niemaksymalnych pikseli)

#### Przykład 1

→ grafika 13.22

#### Przykład 2

→ grafika 13.23

#### Przykład 3

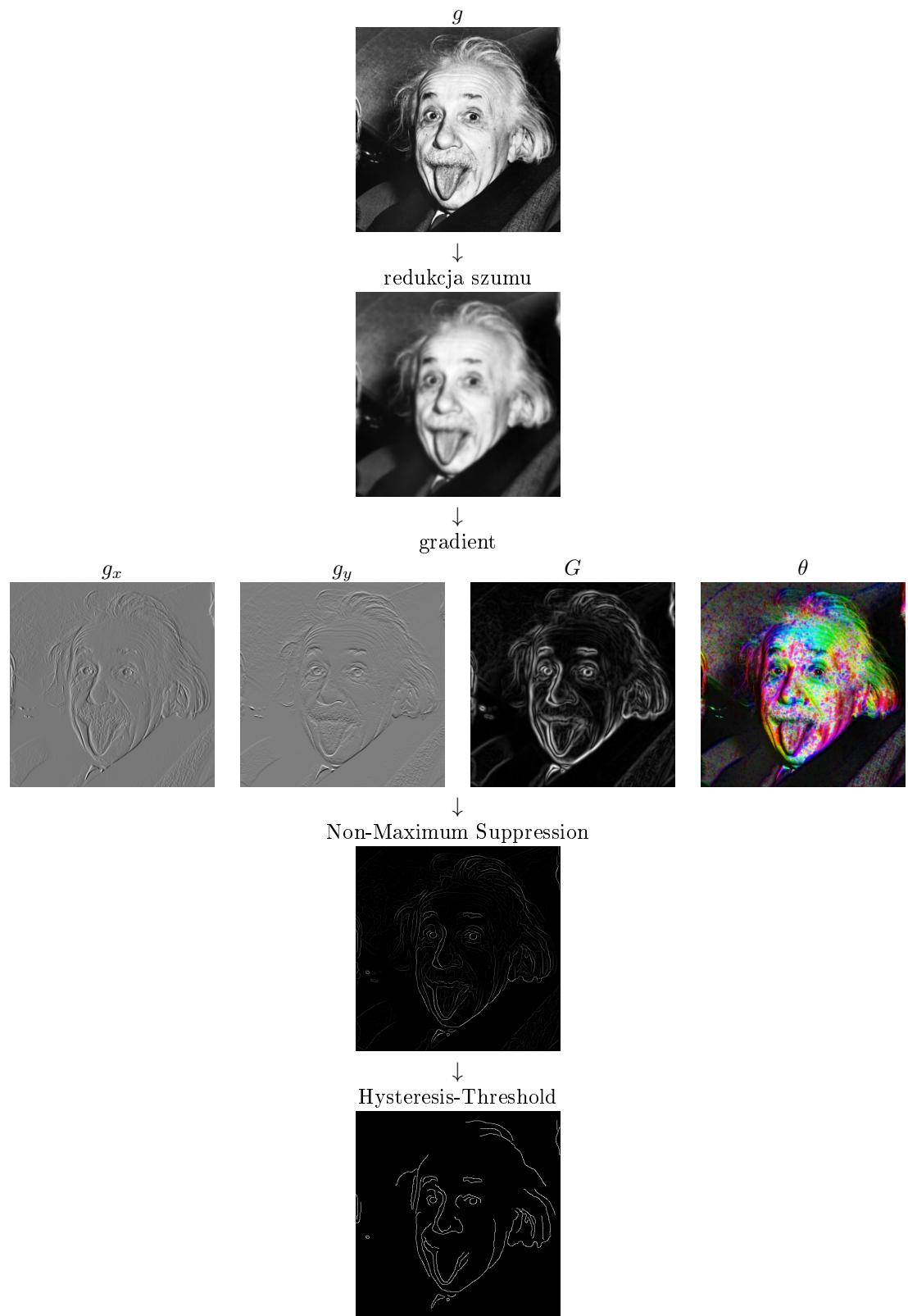
→ grafika 13.24

Zalecana w literaturze kolejność kroków algorytmu i wybór operatora:

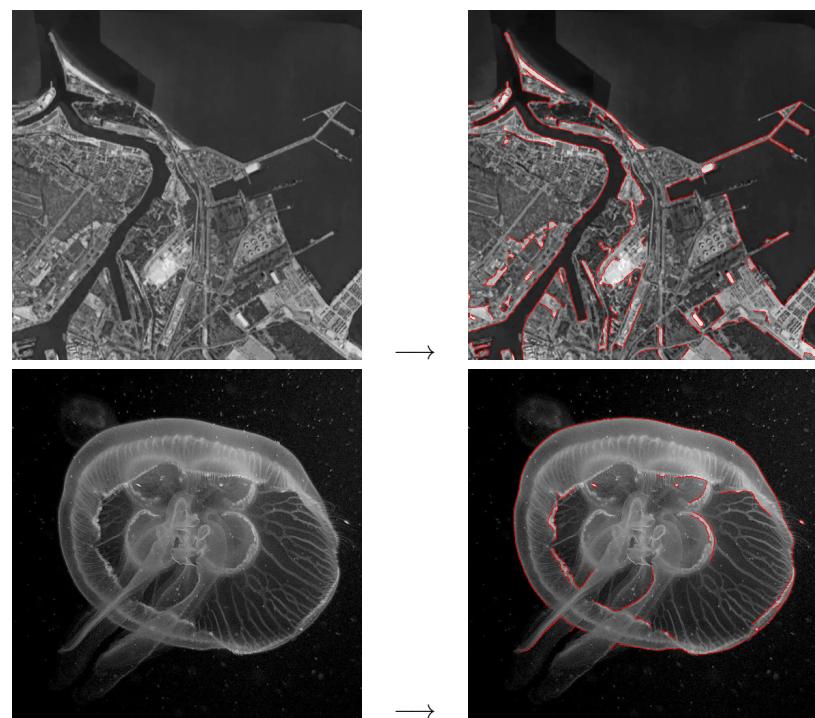
- obliczenie gradientu niezależnie dla każdego kanału,
- wybór operatora zgodnie z normą  $L_\infty$  (maksimum wartości).

**Wyniki:**

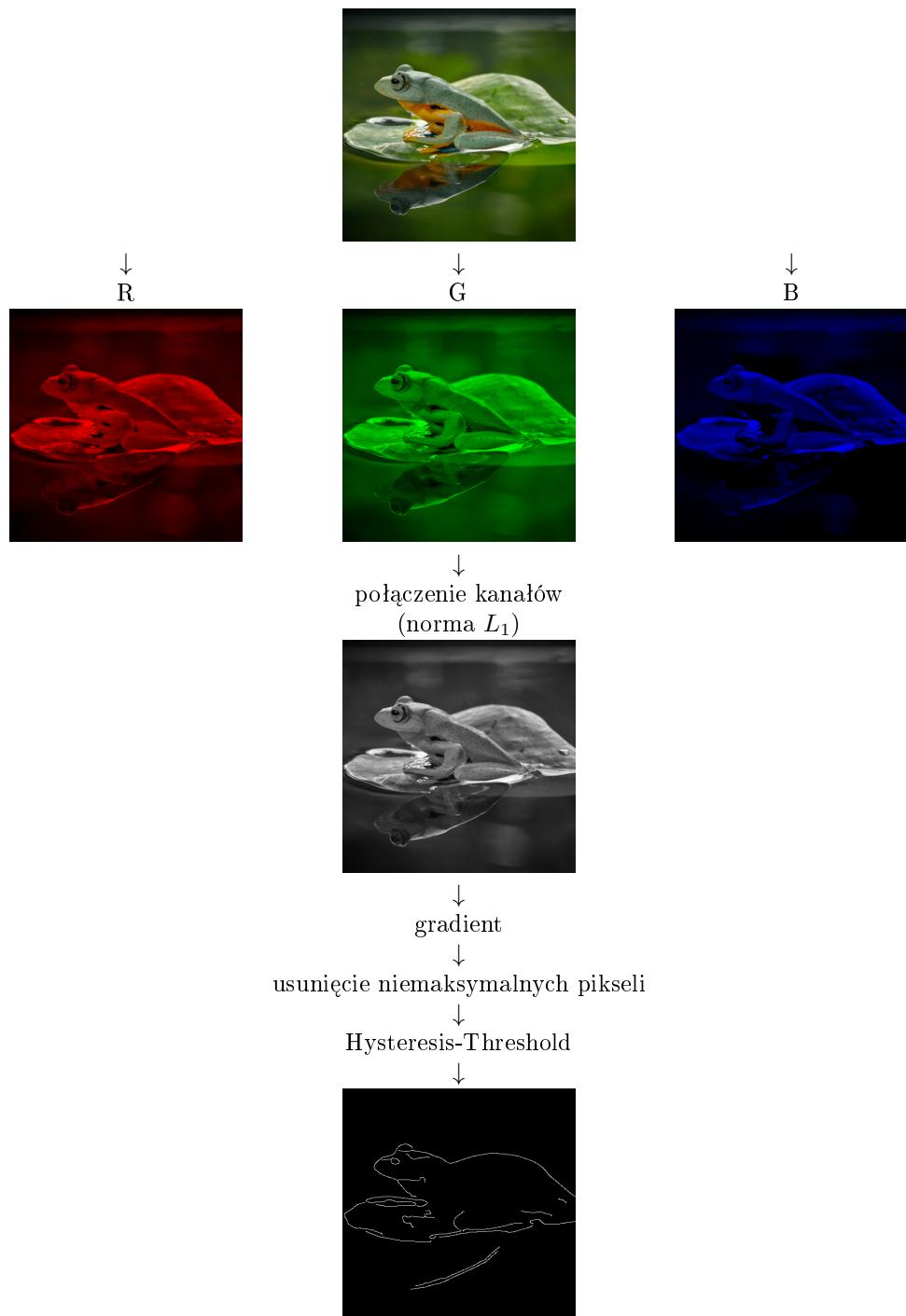
- krawędzie RGB są wyraźniejsze niż dla obrazów w odcieniach szarości,
- większość krawędzi (ponad 90%) jest identyczna.



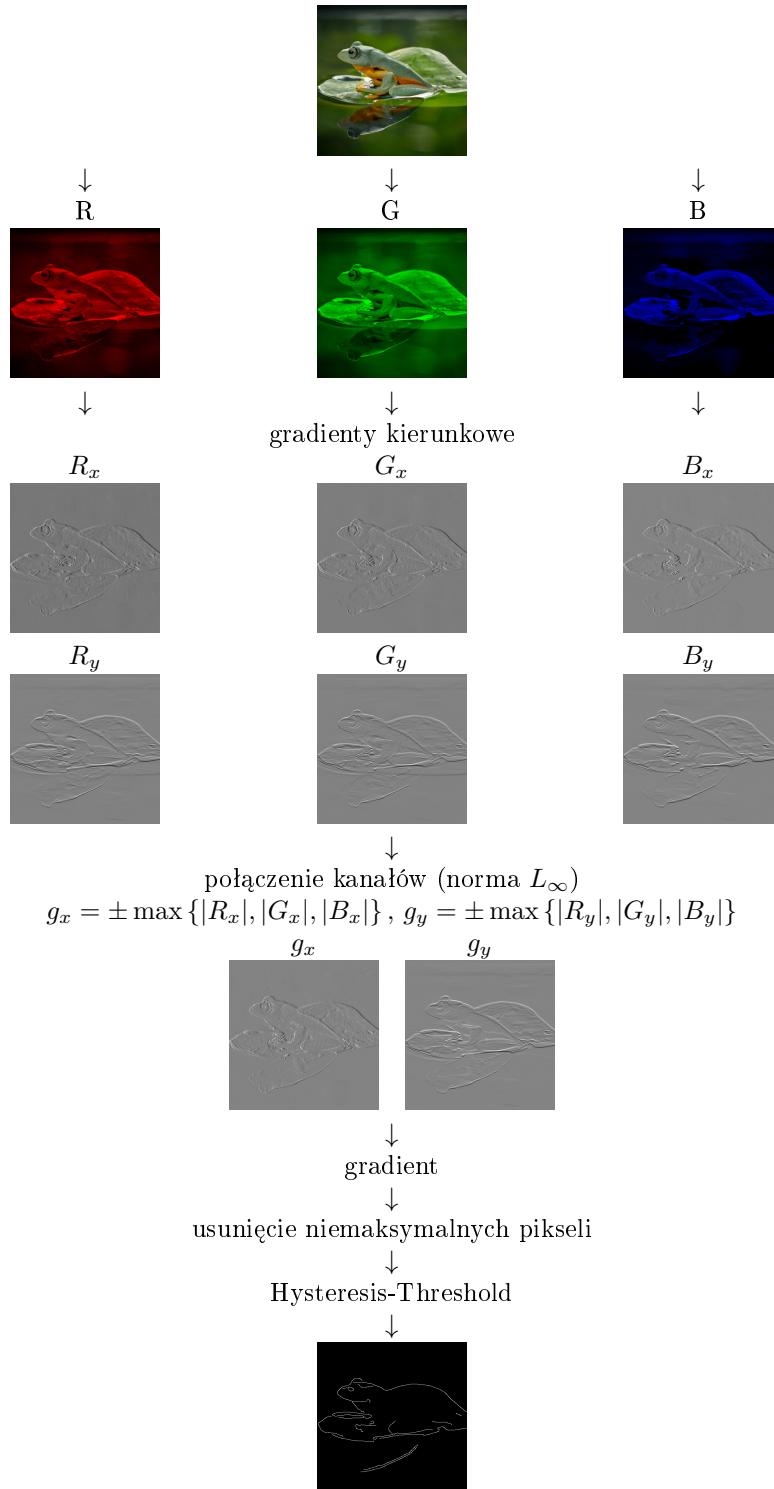
Grafika 13.20: Algorytm wykrywania krawędzi Canny'ego.



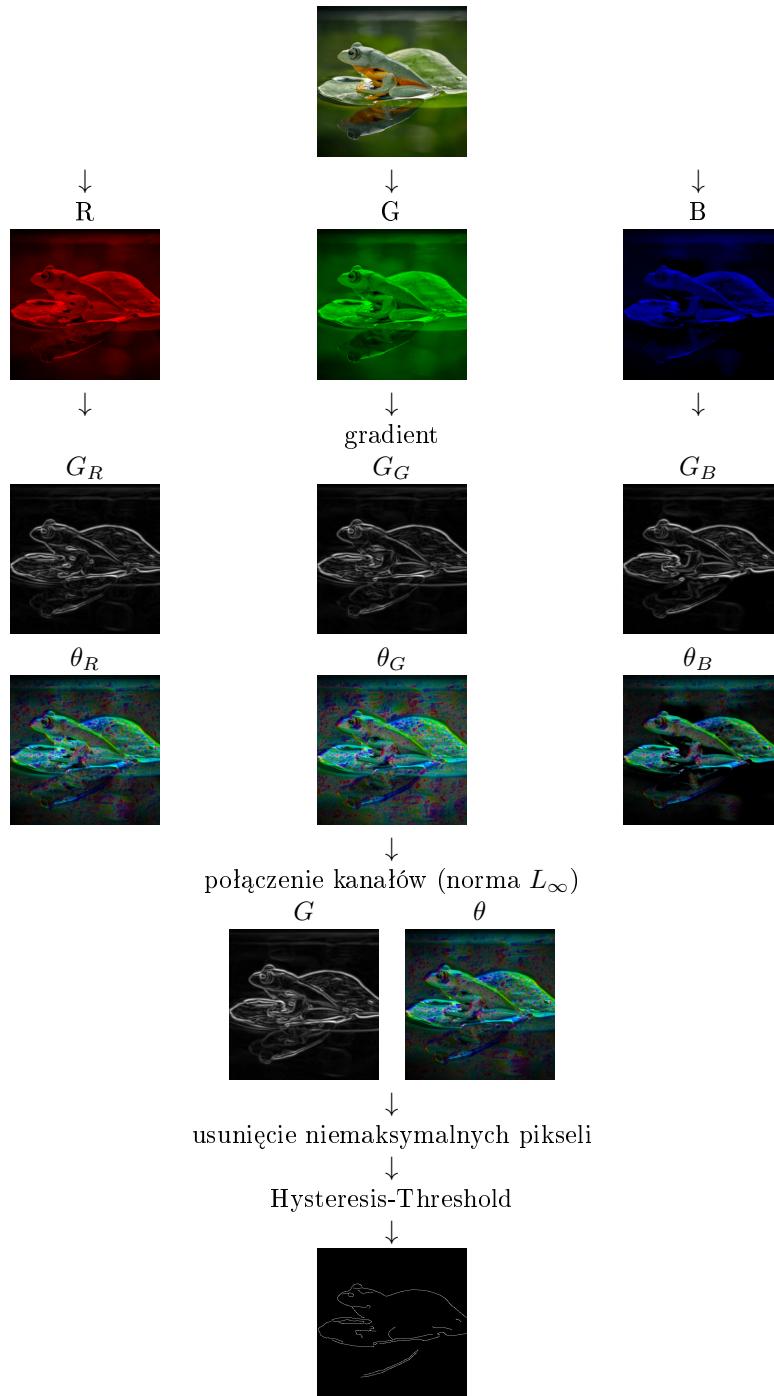
Grafika 13.21: Detektor krawędzi Canny'ego- problemy.



Grafika 13.22: Algorytm wykrywania krawędzi Canny'ego dla obrazów RGB.



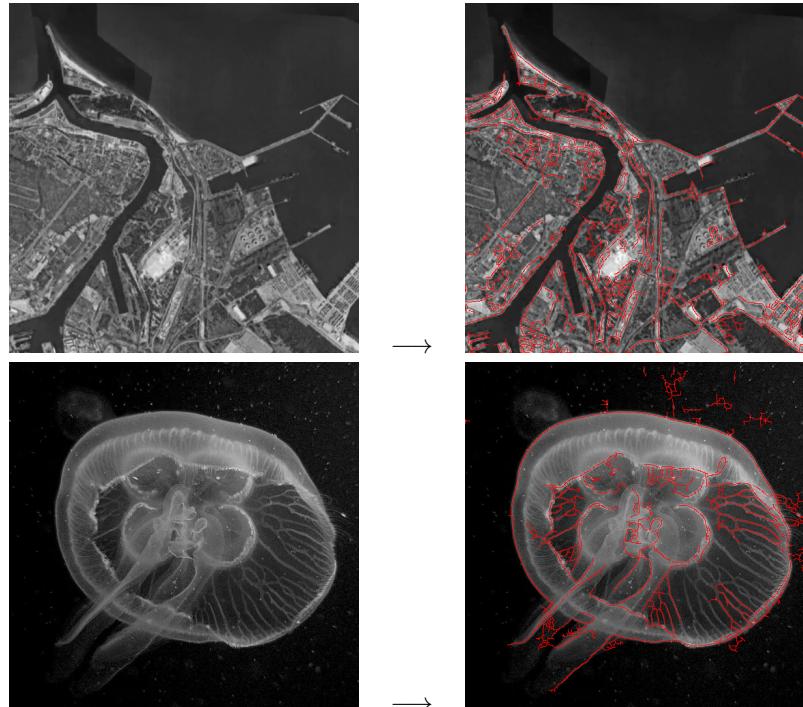
Grafika 13.23: Algorytm wykrywania krawędzi Canny'ego dla obrazów RGB.



Grafika 13.24: Algorytm wykrywania krawędzi Canny'ego dla obrazów RGB.

### 13.5 Detekcja grzbietów (*Ridge Detection*)

→ grafika 13.25



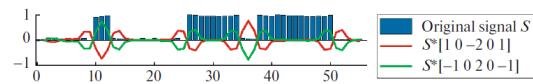
Grafika 13.25: Detektor grzbietów w obrazie

Klasyczne techniki wykrywania krawędzi są zoptymalizowane pod kątem wyodrębniania krawędzi w formie stopni lub ramp; problemem jest wykrycie grzbietów oraz linii "grzebienia" na obrazach (*ridges, crest lines, roof edges* - cienkie siatki wewnętrz obrazu; obrazujące np. drogi czy rzeki na zdjęciach satelitarnych, naczynia krwionośne na zdjęciach medycznych czy korzenie roślin). Funkcje ekstrakcji krawędzi zwracają dwie krawędzie po obu stronach linii grzbietu, ponieważ wąskie grzbiety lub doliny na powierzchni obrazu składają się z dwóch lokalnie równoległych krawędzi.

→ grafika 13.26

W filtrowaniu obrazu do określenia położenia grzbietów można zastosować pochodną drugiego rzędu. Jasne lub ciemne prążki odpowiadają odpowiednio maksimum lub minimum intensywności obrazu w kierunku do nich prostopadłym i stałą intensywności obrazu w kierunku do nich równoległym.

- Do określenia położenia grzbietów można zastosować pochodną drugiego rzędu.
- Operatory LoG i DoG umożliwiają wyodrębnianie grzbietów i dolin, ale nie zawsze są skuteczne, gdy obiekty są zbyt cienkie, przez co wykrywanie jest zakłócone przez szum lub niepożądane artefakty.



Ridge and Valley highlighted on a discrete 1D signal using Laplacian.

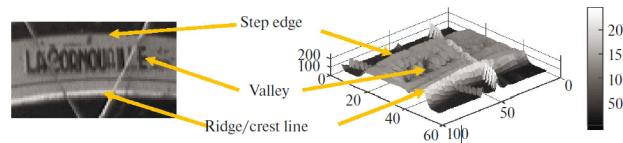


Illustration of features in images by elevation of the image intensity.

Grafika 13.26: Linie grzbietów w obrazie. Źródło: Ghulam-Sakhi Shokouh et al. "Ridge Detection by Image Filtering Techniques" A Review and an Objective Analysis

- Można poprawić selektywność kątową obu operatorów przez zastosowanie algorytmów wykrywania grzbietów.

### 13.5.1 Wykrywanie grzbietów z macierzą Hessego

#### Macierz Hessego, Hesjan

→ macierz kwadratowa, będąca analogiem drugiej pochodnej funkcji w wielowymiarowej analizie rzeczywistej

Dla dwukrotnie różniczkowalnej funkcji  $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$  macierz Hessego  $\mathcal{H}$  w punkcie  $x = (x_1, x_2, \dots, x_n)$  jest zdefiniowana jako

$$\mathcal{H}_f(x) := \left( \frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right)_{i,j=1,\dots,n} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{pmatrix}$$

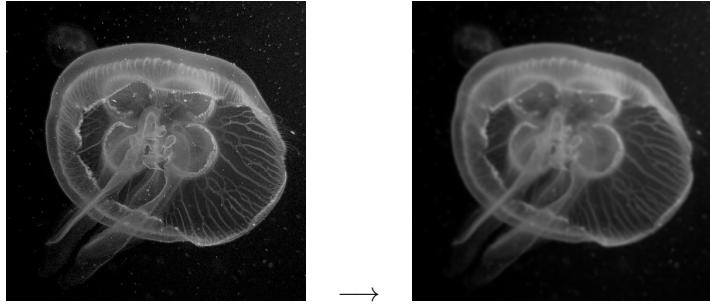
**Zastosowanie:** ekstrema funkcji wielu zmiennych

#### 1. Wygładzenie obrazu

filtr Gausa → grafika 13.27

#### 2. Obliczenie drugich pochodnych kierunkowych obrazu

$$\begin{aligned} g_{xx} &= h_{xx} * g \\ g_{yy} &= h_{yy} * g \\ g_{xy} &= h_{xy} * g \end{aligned}$$



Grafika 13.27: Wygładzenie obrazu (filtr Gaussa)

gdzie pierwsze pochodne kierunkowe  $h_{i,i \in \{x,y\}}$  można (przykładowo) obliczyć ze wzoru

$$h_x = \pm(-1 \ 0 \ 1) \text{ i } h_y = \pm \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix},$$

a drugie pochodne kierunkowe  $h_{ij \ i,j \in \{x,y\}}$  to splot  $h_i * h_j$ :

$$\begin{aligned} h_{xx} &= h_x * h_x = (-1 \ 0 \ 2 \ 0 \ -1) \\ h_{yy} &= h_y * h_y \\ h_{xy} &= h_x * h_y \end{aligned}$$

→ grafika 13.28

### 3. Utworzenie macierzy Hessego i wyznaczenie wartości własnych macierzy

$$\mathcal{H}(m, n) = \begin{pmatrix} g_{xx}(m, n) & g_{xy}(m, n) \\ g_{xy}(m, n) & g_{yy}(m, n) \end{pmatrix} = \begin{pmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ \mathcal{H}_{21} & \mathcal{H}_{22} \end{pmatrix}$$

Macierz jest symetryczna, a jej diagonalizacja wyznacza lokalną normalną do grzbietu lub doliny (czyli wektor własny powiązany z największą wartością własną) i jej ostrość (która jest związana z wartościami tych wartości własnych).

Wartości własne  $k_1, k_2$  macierzy można obliczyć przez:

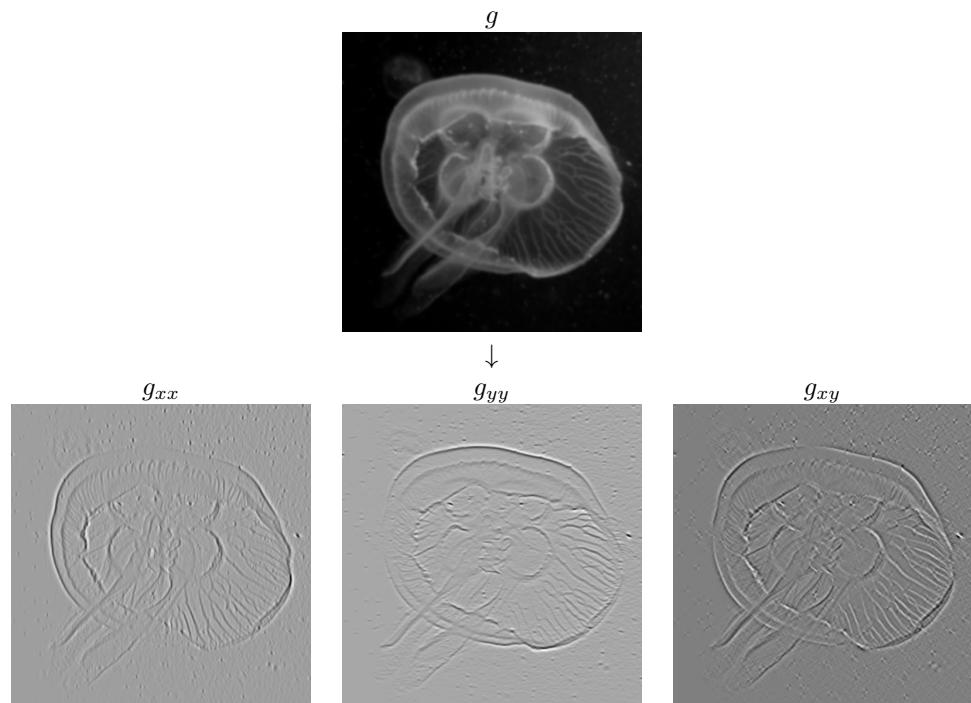
$$\begin{cases} k_1(m, n) = \frac{1}{2}(\mathcal{H}_{11} + \mathcal{H}_{22}) - \frac{1}{4}\sqrt{(\mathcal{H}_{11} + \mathcal{H}_{22})^2 + 4\mathcal{H}_{12}^2} \\ k_2(m, n) = \frac{1}{2}(\mathcal{H}_{11} + \mathcal{H}_{22}) + \frac{1}{4}\sqrt{(\mathcal{H}_{11} + \mathcal{H}_{22})^2 + 4\mathcal{H}_{12}^2} \end{cases}$$

→ grafika 13.29

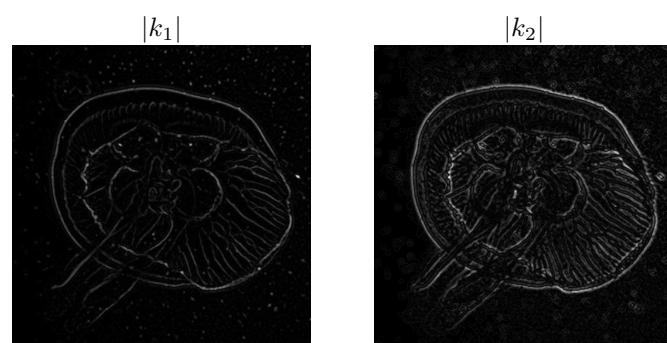
### 4. Wyznaczenie krzywizny obrazu

Dwie wartości własne  $k_1$  i  $k_2$  odpowiadają dwóm głównym krzywiznom lokalnej powierzchni obrazu. Obok nich definiuje się kilka funkcji wskazujących lokalny kontrast obrazu:

- $D_1 = k_1$  (główna wartość własna)



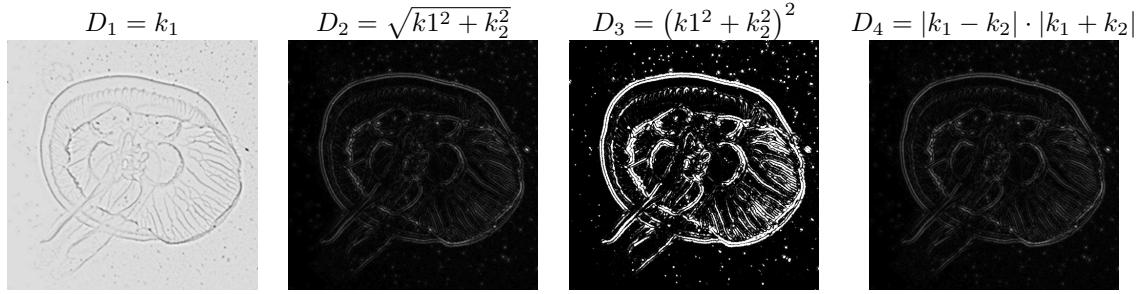
Grafika 13.28: Drugie pochodne kierunkowe wygładzonego obrazu



Grafika 13.29: Wartości własne  $k_1, k_2$  macierzy Hessego.

- $D_2 = \sqrt{k_1^2 + k_2^2}$
- $D_3 = (k_1^2 + k_2^2)^2$
- $D_4 = |k_1 - k_2| \cdot |k_1 + k_2|$

→ grafika 13.30



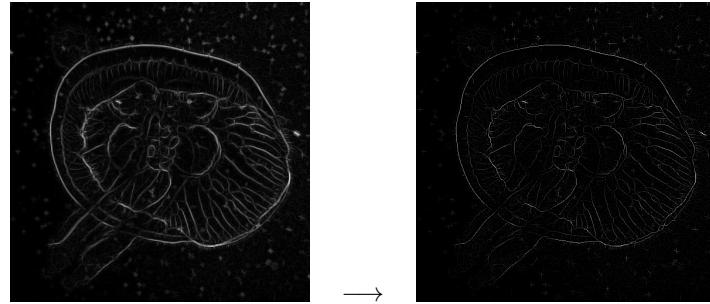
Grafika 13.30: Wyznaczenie krzywizny obrazu z wartością własnych  $k_1$  i  $k_2$  macierzy Hessego.

### 5. Non-Maximum Suppression

Ewentualne usunięcie niemaksymalnych pikseli:  
piksel jest oznaczony jako piksel grzbietu/doliny, jeżeli  $D_{i,i \in \{1,2,3,4\}}$  jest maksymalne w kierunku danym przez wektor własny (kierunek prostopadły do grzbietu/doliny):

$$\theta = \begin{pmatrix} \mathcal{H}_{12} \\ k_1 - \mathcal{H}_{11} \end{pmatrix}$$

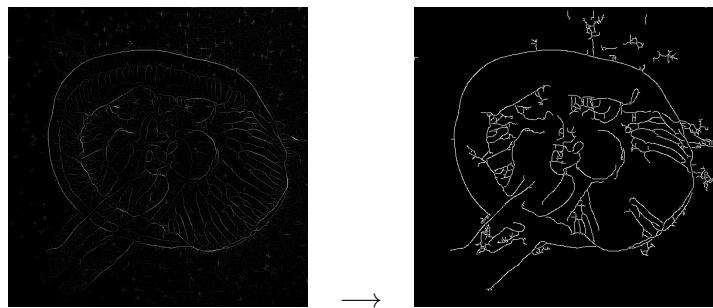
→ grafika 13.31



Grafika 13.31: Non-Maximum Suppression (usunięcie niemaksymalnych pikseli)

### 6. Hysteresis threshold (progowanie histerezy)

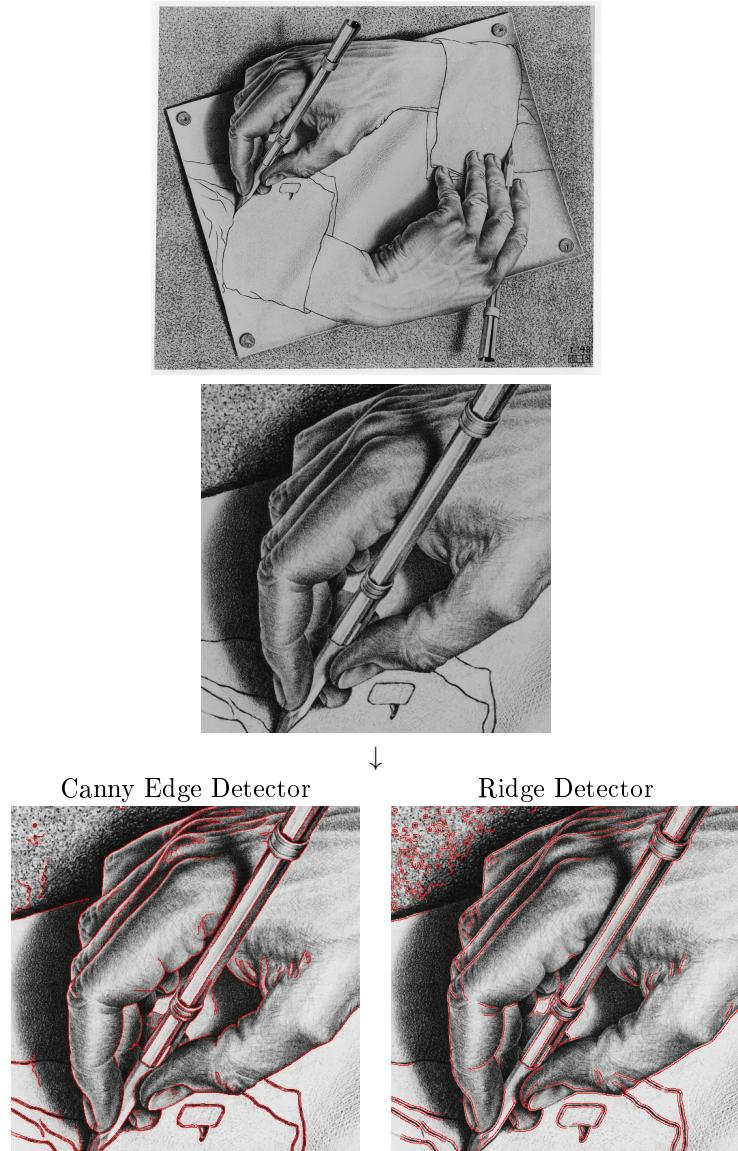
→ grafika 13.32



Grafika 13.32: Hysteresis threshold (progowanie histerezy)

**Porównanie Canny Edge Detector - Ridge Detector**

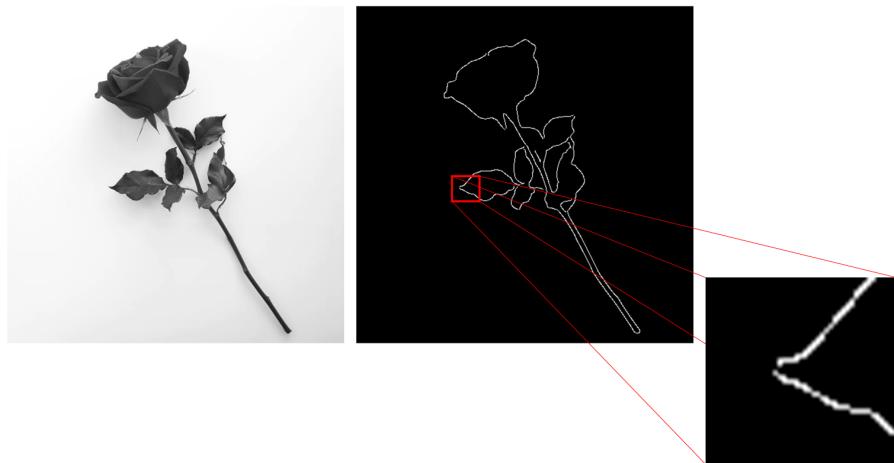
→ grafika 13.33



Grafika 13.33: Wykrywanie krawędzi i grzbietów w obrazie. Obraz wejściowy: M.C. Escher, Drawing Hands. ©2017 The M.C. Escher Company, The Netherlands.

### 13.6 Detekcja katów obiektów (*Corner detection*)

→ grafika 13.34



Grafika 13.34: Detekcja krawędzi algorytmem Canny'ego - problem z lokalizacją narożnika obiektu.

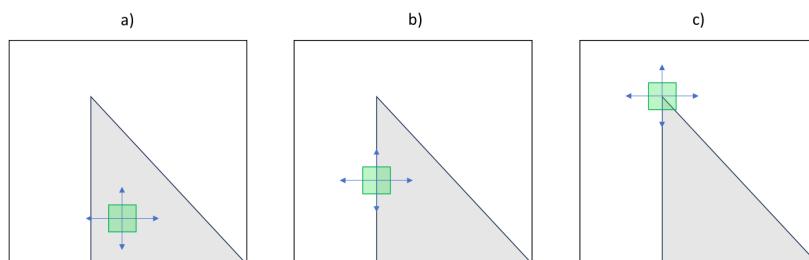
Kąt (narożnik) obiektu

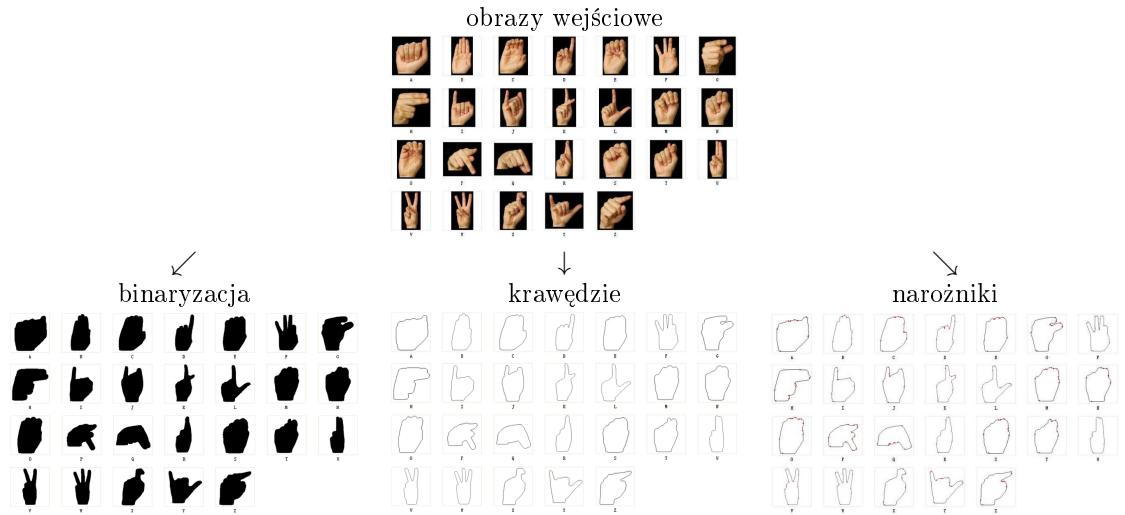
- przecięcie dwóch krawędzi
  - punkt, dla którego w lokalnym sąsiedztwie występują dwa dominujące i różne kierunki krawędzi.

**Przykład:** Rozpoznawanie gestów w języku migowym → grafika 13.35

### 13.6.1 Harris Corner Detection

Motywacja

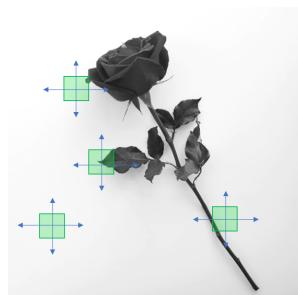




Grafika 13.35: 26 przykładowych obrazów gestów dloni dla 26 liter w amerykańskim języku migowym. Zródło: Linghua Li "Corner Detection of Hand Gesture" TELKOM-NIKA Indonesian Journal of Electrical Engineering 10(8)

- a) obszar bez zmian gradientu
- b) gradient nie zmienia się w kierunku krawędzi → krawędź
- c) gradient zmienia się we wszystkich kierunkach → narożnik

→ Wokół każdego piksela  $p$  obrazu rozważane jest małe okno, które jest przesuwane o niewielką wartość w danym kierunku. Mierzona jest wielkość zmiany zachodzącej w wartościach pikseli (gradient).



### Przesunięcie okna

Efekt przesunięcia okna zawierającego piksele o współrzędnych  $(x, y) \in \text{window}$  o (małe) wartości  $\Delta x, \Delta y$ :

- różnica intensywności  $s(\Delta x, \Delta y)$  (suma kwadratów różnicy (*sum squared difference, SSD*) wartości pikseli przed i po przesunięciu okna):

$$s(\Delta x, \Delta y) = \sum_{(x,y) \in \text{window}} [g(x, y) - g(x + \Delta x, y + \Delta y)]^2$$

- liniarne przybliżenie dla małych wartości  $\Delta x, \Delta y$

$$g(x + \Delta x, y + \Delta y) \approx g(x, y) + g_x(x, y)\Delta x + g_y(x, y)\Delta y$$

$g_x, g_y$  to gradienty w kierunkach  $x$  i  $y$

$$\begin{aligned} s(\Delta x, \Delta y) &\approx \sum_{(x,y) \in \text{window}} [g_x(x, y)\Delta x + g_y(x, y)\Delta y]^2 \\ &= \sum_{(x,y) \in \text{window}} [(g_x(x, y) \ g_y(x, y)) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}]^2 \\ &= \sum_{(x,y) \in \text{window}} [(g_x(x, y) \ g_y(x, y)) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}] \left[ (g_x(x, y) \ g_y(x, y)) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right] \\ &= \sum_{(x,y) \in \text{window}} \left[ (\Delta x \ \Delta y) \begin{pmatrix} g_x(x, y) \\ g_y(x, y) \end{pmatrix} \right] \left[ (g_x(x, y) \ g_y(x, y)) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right] \\ &= (\Delta x \ \Delta y) \left[ \sum_{(x,y) \in \text{window}} \begin{pmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{pmatrix} \right] \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \\ &= (\Delta x \ \Delta y) \underbrace{\left( \begin{array}{cc} \sum_{(x,y) \in \text{window}} g_x^2 & \sum_{(x,y) \in \text{window}} g_x g_y \\ \sum_{(x,y) \in \text{window}} g_x g_y & \sum_{(x,y) \in \text{window}} g_y^2 \end{array} \right)}_{\mathcal{M}} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \\ &= (\Delta x \ \Delta y) \mathcal{M} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \end{aligned}$$

#### *Second-moment matrix*

- Macierz  $\mathcal{M}$  oblicza się sumując pochodne kierunkowe funkcji obrazu  $g$  w obszarze  $(x, y) \in \text{window}$  wokół punktu  $(x, y)$ :

$$\mathcal{M} = \sum_{(x,y) \in \text{window}} \begin{pmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{pmatrix}$$

- $\mathcal{M}$  opisuje strukturę sąsiedztwa wokół lokalizacji  $(x, y) \rightarrow$  jej rząd różni się w zależności od cech środowiska:

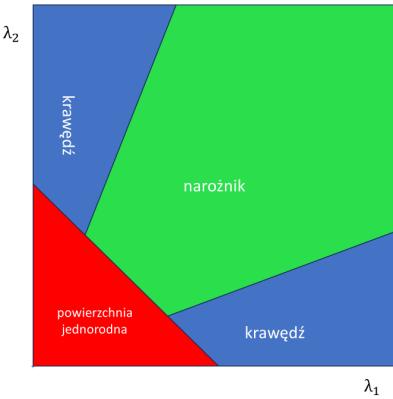
- $\text{rang}(\mathcal{M}) = 2$  punkt narożnikowy
- $\text{rang}(\mathcal{M}) = 1$  (prosta) krawędź
- $\text{rang}(\mathcal{M}) = 0$  jednorodna, niestrukturyzowana powierzchnia

- Wartości własne  $\lambda_1, \lambda_2$  macierzy  $\mathcal{M}$  opisują struktury sąsiedztwa:  $\lambda_1, \lambda_2$  są proporcjonalne do zmian wartości szarości w obrazie wzdłuż głównych kierunków. Dzięki tym właściwościom wartości własne idealnie nadają się do oceny struktury sąsiedztwa.

Analiza przestrzeni parametrów prowadzi do trzech przypadków, które można wyróżnić:

- a) Jeśli obie wartości własne  $\lambda_1, \lambda_2$  są małe, wówczas zmiany wartości szarości w głównych kierunkach są również małe, tj. wartości szarości są stałe w środowisku  $\rightarrow$  brak krawędzi i narożnika.

- b) Jeżeli jedna wartość własna jest duża, a druga mała, to następuje lokalna zmiana szarości w jednym kierunku. Duża wartość własna wskazuje na dużą zmianę wartości szarości prostopadłe do krawędzi, podczas gdy mała wartość własna wzdłuż krawędzi oznacza brak lub jedynie niewielką zmianę wartości szarości  $\rightarrow$  krawędź.
- c) Jeżeli obie wartości własne są duże, tj. zmiany wartości szarości w obu kierunkach są również duże, jest to punkt narożny.



- Aby móc przeprowadzić klasyfikację w celu rozróżnienia przypadków od a) do c), potrzebna jest funkcja oparta na wartościach własnych, która pokazuje "siłę" punktu narożnika. By uniknąć obliczania wartości własnych  $\lambda_1, \lambda_2$  macierzy  $\mathcal{M}$ , stosuje się następujące zależności dla wyznacznika ( $\det$ ) i śladu macierzy ( $\text{tr}$ ):

$$\begin{aligned}\det\mathcal{M} &= \lambda_1 \cdot \lambda_2 \\ \text{tr}\mathcal{M} &= \lambda_1 + \lambda_2\end{aligned}$$

#### Miara "siły" narożników

$$\begin{aligned}R &= \lambda_1 \cdot \lambda_2 - k(\lambda_1 + \lambda_2)^2 \\ &= \det\mathcal{M} - k(\text{tr}\mathcal{M})^2\end{aligned}$$

gdzie

$$k = 0.04 \dots 0.06$$

#### Wykrywanie narożników obiektów – algorytm Harrisa

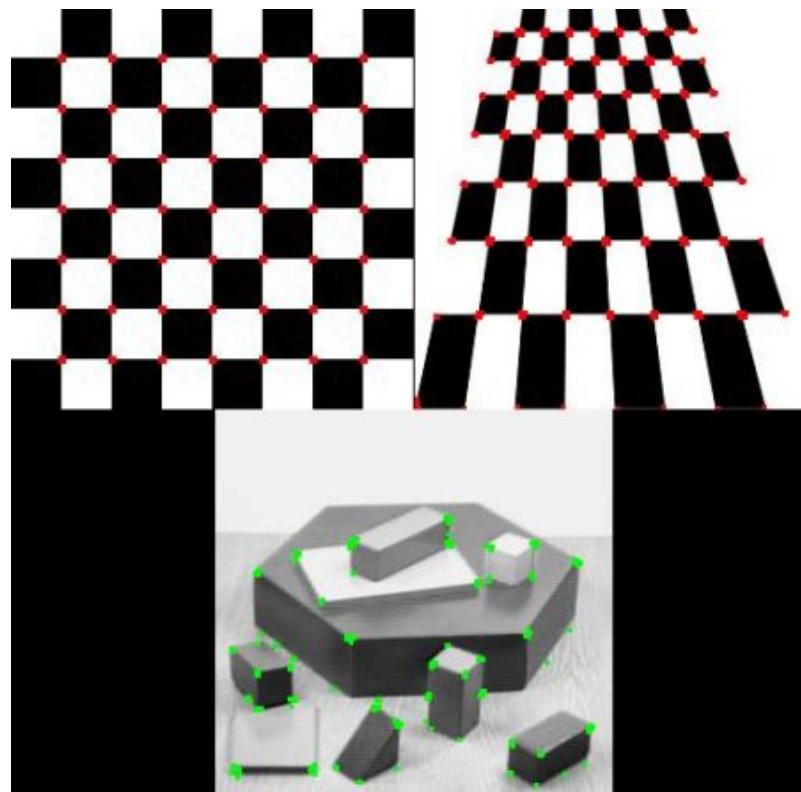
1. obliczenie pochodnych kierunkowych  $g_x$  i  $g_y$  obrazu  
 $\rightarrow$  filtry gradientowe, operator Sobel'a,...
2. obliczenie iloczynów pochodnych dla każdego piksela

$$g_x^2 = g_x g_x, \quad g_y^2 = g_y g_y, \quad g_{xy} = g_x g_y$$

3. dla każdego piksela  $(m, n)$  obliczenie sumy w oknie go otaczającym  $(x, y) \in \text{window}$

4. zdefiniowanie dla każdego piksela  $(m, n)$  macierzy  $\mathcal{M}$
5. obliczenie miary siły narożnika dla każdego piksela  $(m, n)$
6. Non-Maximum Suppression (usunięcie niemaksymalnych pikseli)

Przykład: → grafika 13.36

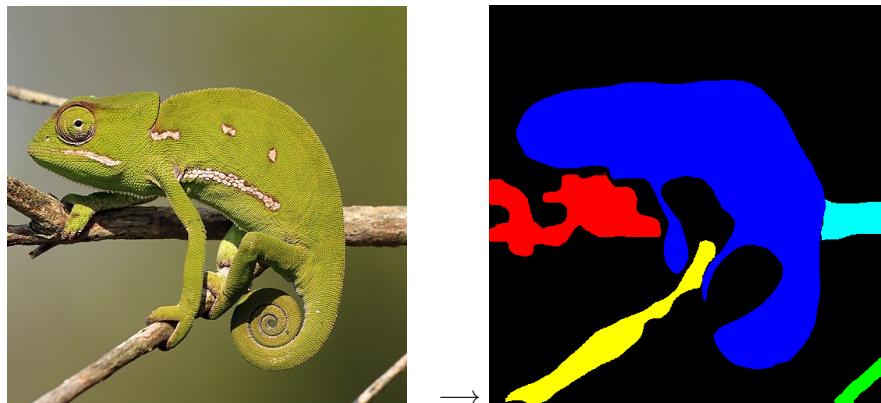


Grafika 13.36: Detekcja narożników algorytmem Harrisa. Źródło: Open Source Computer Vision, Harris Corner Detection

# 14

## Segmentacja obiektów

→ grafika 14.1



Grafika 14.1: Segmentacja obrazu

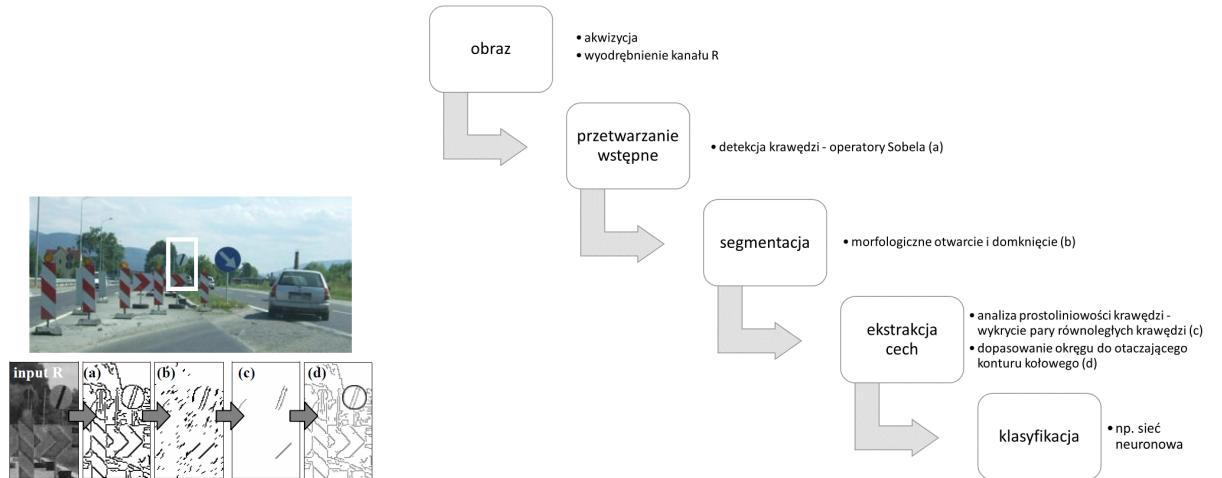
Celem segmentacji obrazu jest podzielenie obrazu na wizualnie różne obszary, które są jednorodne pod względem pewnych właściwości (np. wartość szarości, kolor, tekstuра).

W procesie widzenia komputerowego segmentacja jest zwykle pierwszym krokiem analizy obrazu i następuje po jego wstępnym przetworzeniu. → grafika 14.2

### 14.0.1 Segmentacja: rozkład obrazu na jednostki semantyczne

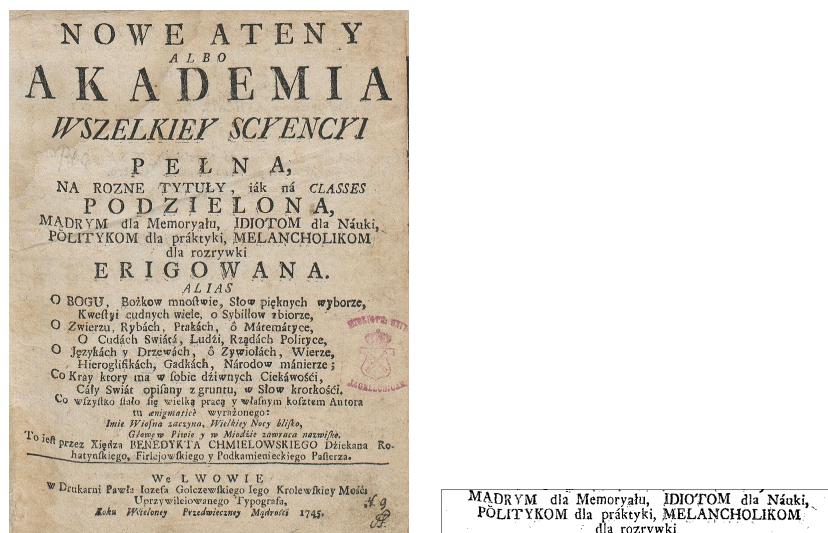
→ grafika 14.3 Pożądane właściwości segmentacji:

- kompletność: każdy piksel jest przypisany do segmentu,
- jednoznaczność: piksel jest przypisany do dokładnie jednego segmentu



Grafika 14.2: Analiza obrazu - detekcja znaków "koniec ograniczenia prędkości". Źródło: "Integrated Speed Limit Detection and Recognition from Real-Time Video" M. Eichner, T.P. Breckon, IEEE Intelligent Vehicles Symposium

- zwartość: każdy segment tworzy połączony obszar



Grafika 14.3: Obraz i jego segment

### Opis matematyczny

Segmentacja obrazu  $g(m, n)$  o wartościach intensywności (szarości)  $\mathcal{G}$  to funkcja

$$A : (m, n) \longrightarrow \{0, 1, 2, \dots\}$$

która przypisuje klasę (lub etykietę) do każdego piksela na podstawie kryteriów decyzyjnych związanych z obrazem.

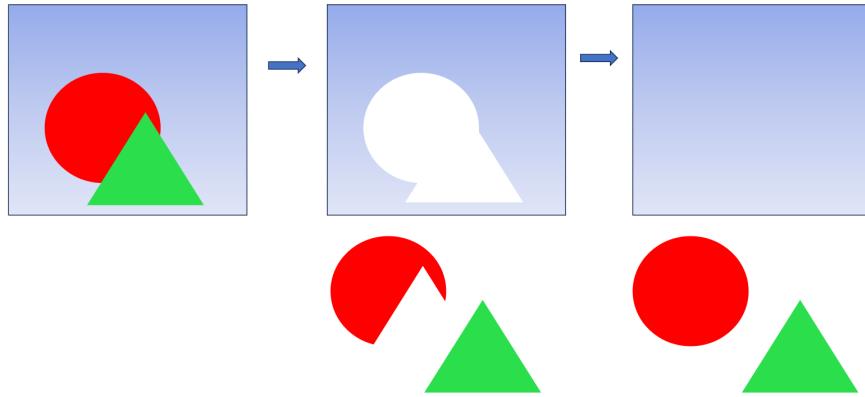
### Przykład

Segmentacja pojedynczego obiektu w obrazie za pomocą wartości etykiety 1:

$$A(m, n) = \begin{cases} 1 & \text{piksel } (m, n) \text{ należy do obiektu} \\ 0 & \text{piksel } (m, n) \text{ nie należy do obiektu} \end{cases}$$

### Interpretacja segmentów

→ grafika 14.4

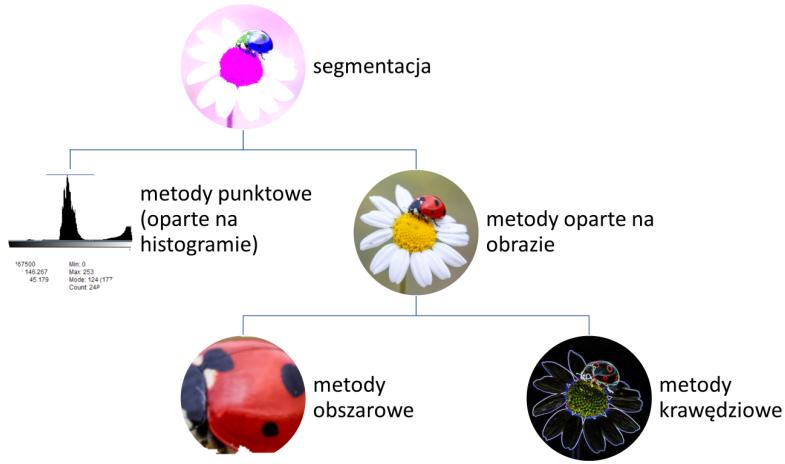


Grafika 14.4: Interpretacja segmentów na obrazie

### Klasyfikacja metod segmentacji

→ grafika 14.5

Przyporządkowanie procedur segmentacji do klasyfikacji jest często niejednoznaczne. Można także łączyć różne procedury, aby osiągnąć lepsze rezultaty.



Grafika 14.5: Klasyfikacja metod segmentacji

## 14.1 Metody punktowe

### 14.1.1 Globalne metody progowania

→ rozdział "Globalne metody progowania"

$$g_T(m, n) = \begin{cases} 0 : & g(m, n) \leq T \\ 1 : & g(m, n) > T \end{cases}$$

Algorytmy wyznaczenia progu  $T$ :

- metoda Otsu
- metoda średniej (ImageJ: *Mean*)  
→ próg jest średnią poziomów szarości
- metoda Kittlera i Illingworth (ImageJ: *MinError(I)*)  
→ przybliżenie histogramu obrazu  $h(h)$  przez dwie funkcje Gaussa:

$$h(g) \approx h_q(g) = a_0 e^{-\frac{(g-\mu_0)^2}{\sigma_0^2}} + a_1 e^{-\frac{(g-\mu_1)^2}{\sigma_1^2}}$$

z zachowaniem warunku:

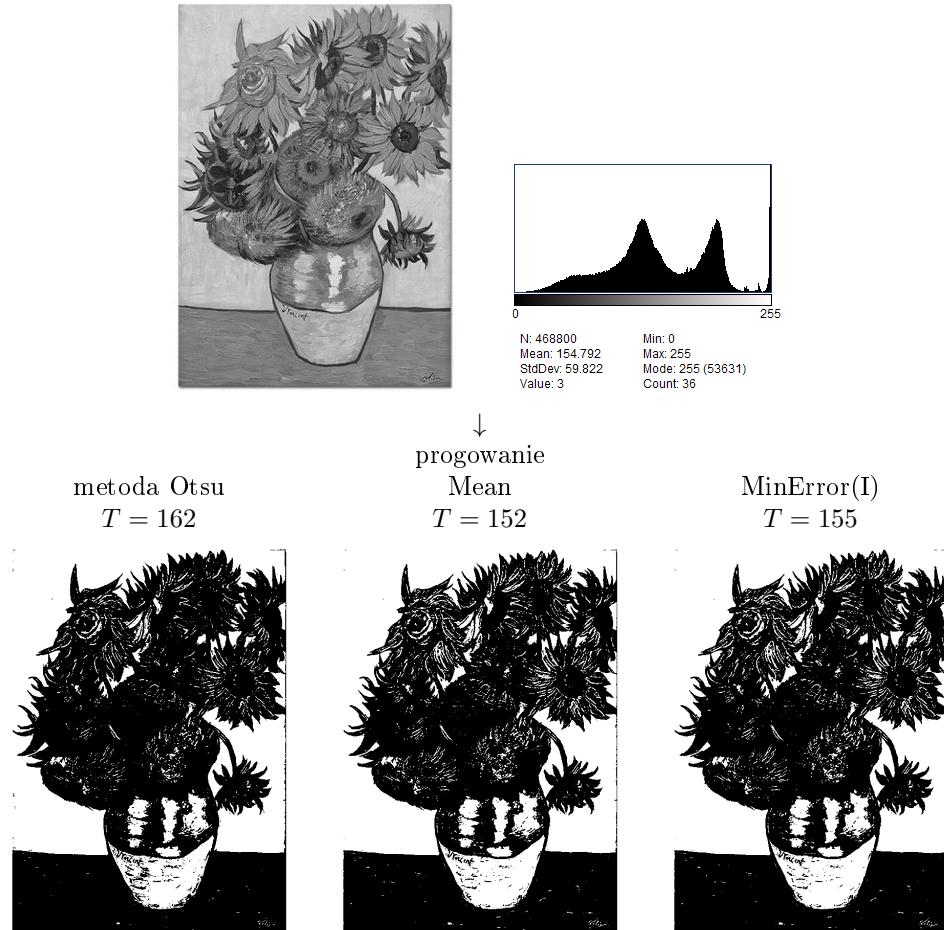
$$\sum_{g=0}^{G-1} |h(g) - h_q(g)| \longrightarrow \min$$

→ przecięcie dwóch krzywych Gaussa wyznacza wartość progową  $T$

• ...

→ grafiki 14.6, 14.7

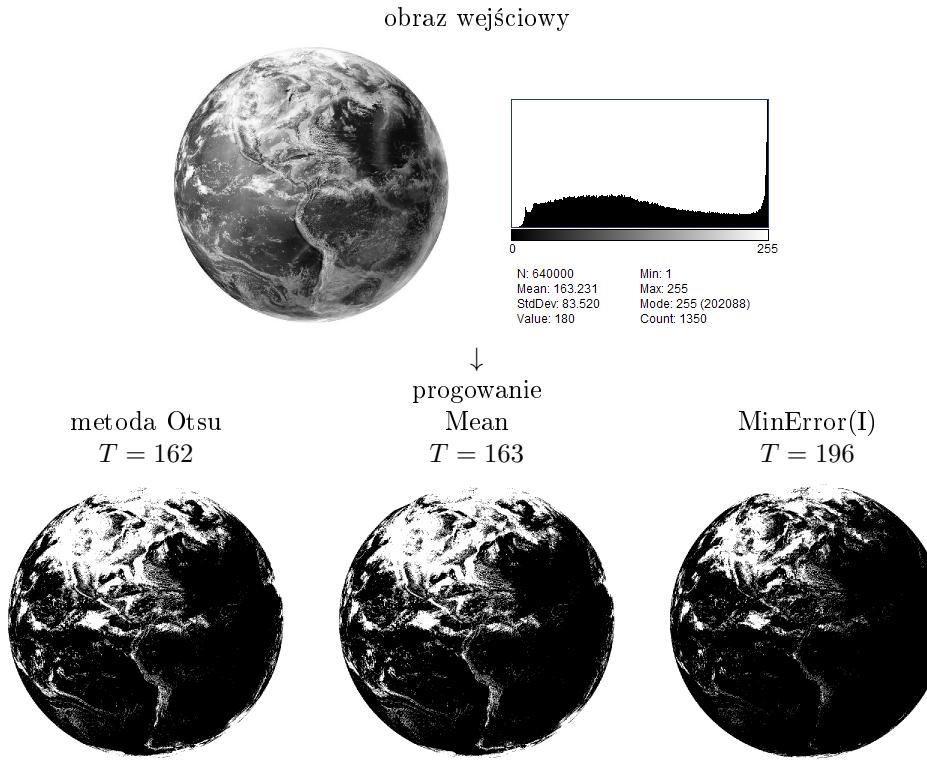
obraz wejściowy



Grafika 14.6: Globalne metody progowania - obraz z histogramem bimodalnym

## Przetwarzanie wstępne: wygładzenie obrazu

Przed użyciem metody wartości progowej warto wygładzić obraz za pomocą filtra. Po wygładzeniu znalezione obszary są bardziej zwarte, a dziury w tych obszarach są częściej wypełnione. Szczególnie odpowiednie są filtry nieliniowe (np. filtry medianowe), ponieważ nie rozmywają krawędzi. → grafika 14.8



Grafika 14.7: Globalne metody progowania - obraz z histogramem jednomodalnym

### Zacienienie *Shading*

Zmiany jasności niszczą bimodalny rozkład częstotliwości.  
→ nie można zdefiniować wartości progowej dla całego obrazu

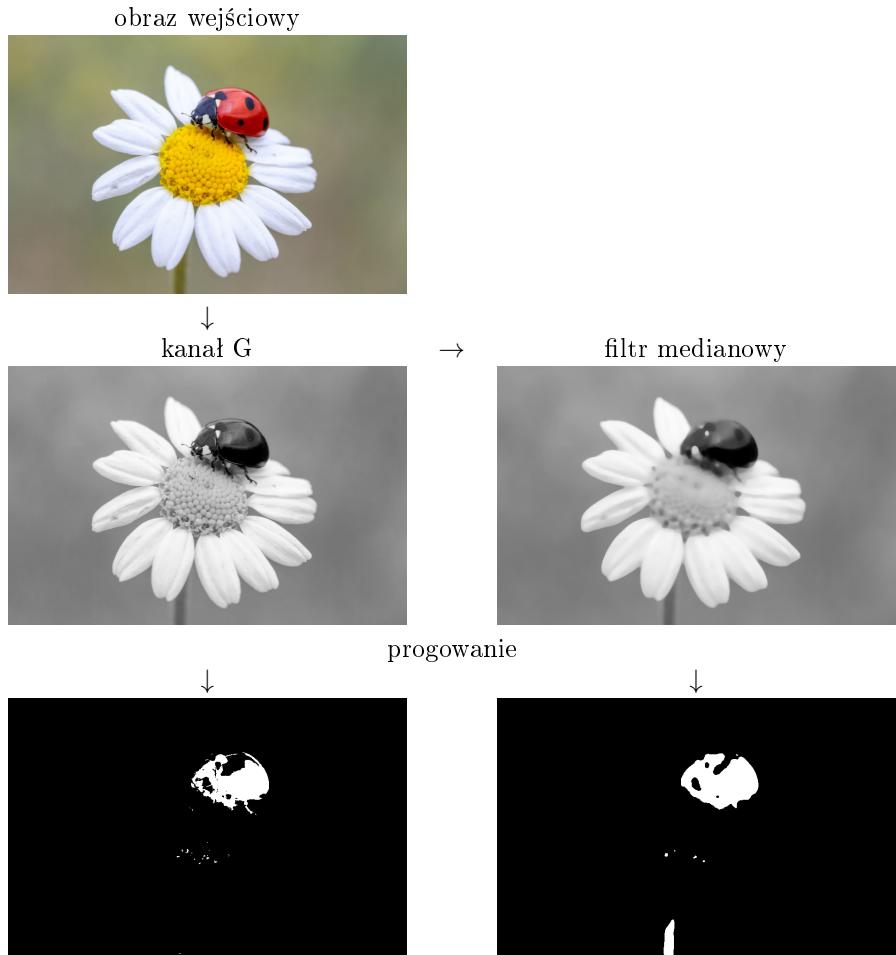
→ grafika 14.9

Uwzględnienie zacienienia:

- jednorodny obraz wykonany w tych samych warunkach
- wyznaczenie obrazu zacienienia na podstawie obrazu wejściowego
- różne (lokalne) progi dla progowania obrazu (progowanie lokalne)

### Przykład: wyznaczanie obrazu zacienienia

- jeżeli zdecydowana większość obrazu składa się z pikseli pierwszego planu lub tła, obraz zacienienia można wygenerować przy użyciu filtra rankingowego



Grafika 14.8: Przetwarzanie wstępne: filtr medianowy (hybrydowa metoda segmentacji: rozdzielenie kanałów + progowanie)

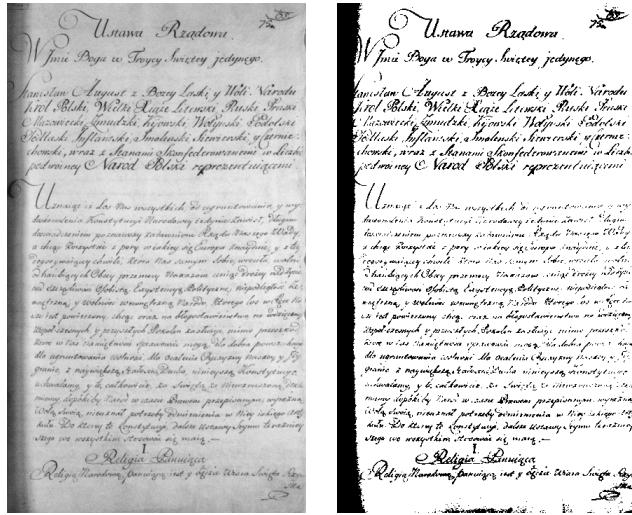
- filtr rankingowy musi być wystarczająco duży, aby zawsze zawierał co najmniej jeden piksel pierwszego planu i jeden piksel tła

→ grafika 14.10

### 14.1.2 Progowanie lokalne

W wielu przypadkach globalny próg dla całego obrazu nie daje dobrych wyników. Lepsze wyniki można osiągnąć stosując lokalne progi → grafika 14.11

- Aby znaleźć lokalne progi, cały obraz dzieli się na mniejsze obszary i z tych obszarów wyznaczane są lokalne histogramy i lokalne wartości progowe.



Grafika 14.9: Shading

- Lokalne metody progowania wyznaczają **próg dynamiczny**.
- W skrajnych przypadkach dla każdego piksela obliczany jest oddzielny próg, który wynika z określonego otoczenia piksela.

### Próg zmienny

- progi lokalne z histogramów w podregionach
- (biliniowa) interpolacja wartości progowych  $T(m, n)$  we wszystkich pozostałych punktach
- segmentacja według  $g(m, n) > T(m, n)$

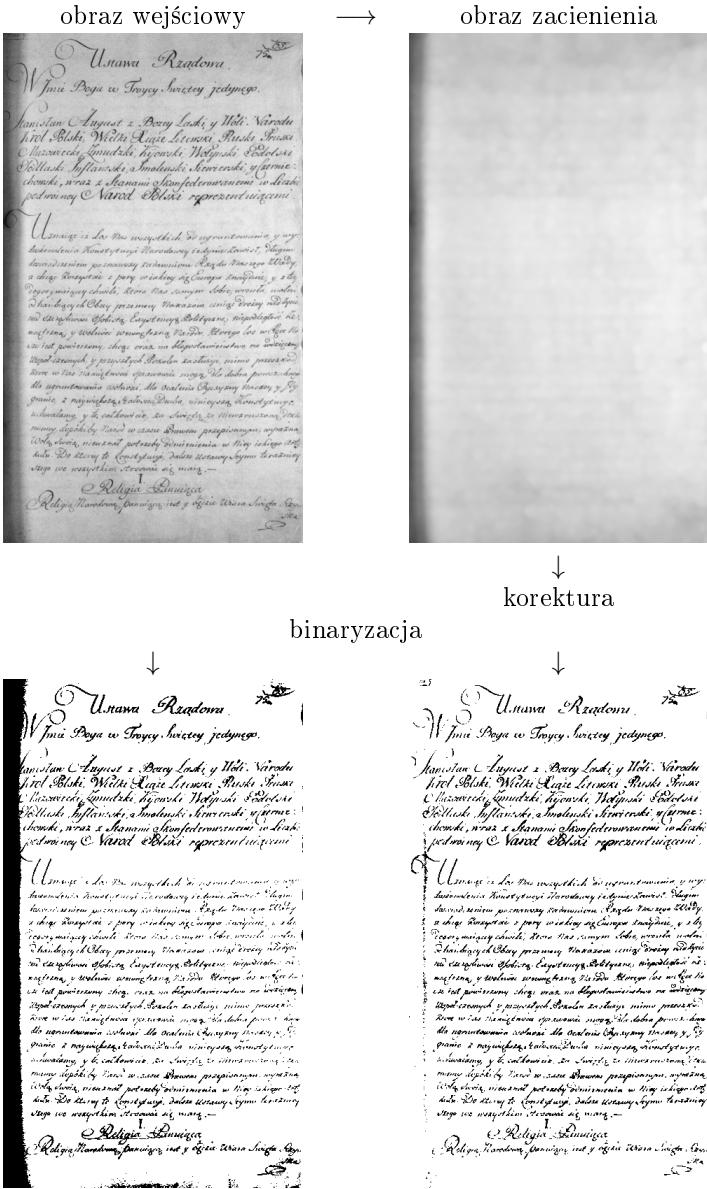
→ grafika 14.12

## 14.2 Metody krawędziowe

W tych metodach wyszukiwane są krawędzie lub przejścia obiektów na obrazie. Wykryte krawędzie wyznaczają granicę obiektu.

→ rozdział "Detekcja krawędzi".

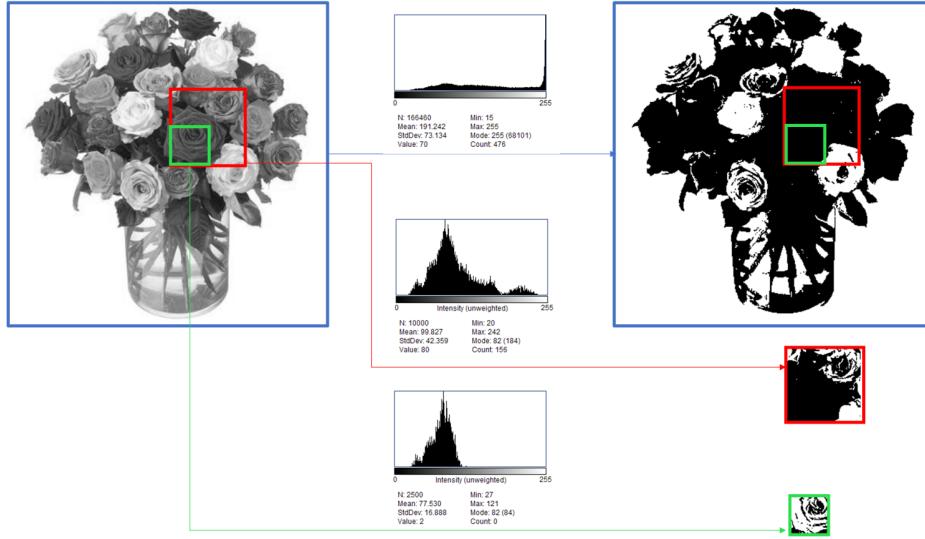
→ grafika 14.13



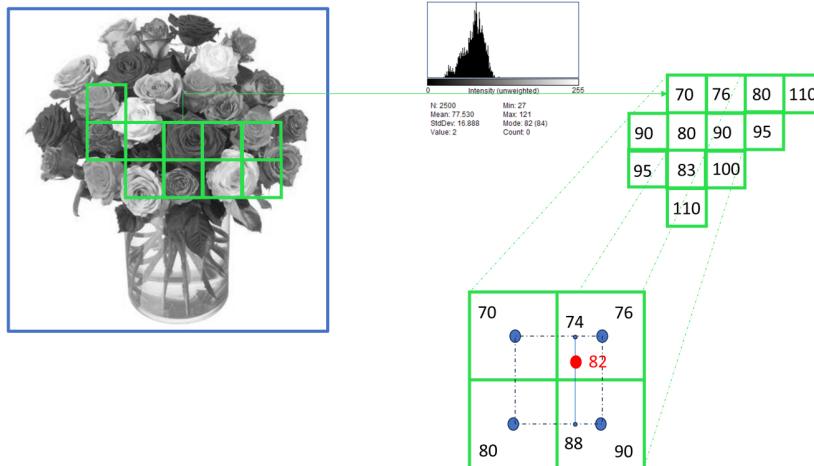
Grafika 14.10: Korektura zacienienia. Obraz zacienienia obliczony przy użyciu filtra rankingowego (maksymalnego).

#### 14.2.1 Transformacja wododziałowa (*Watershed Transformation, WTH*)

→ rozdział "Operatory morfologiczne"  
→ grafika 14.14



Grafika 14.11: Progowanie globalne i lokalne

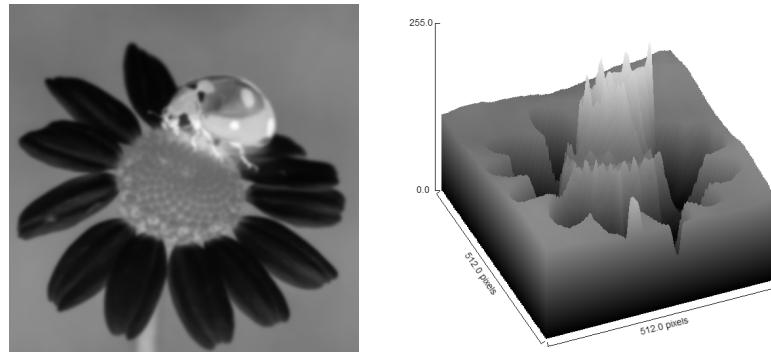


Grafika 14.12: Próg zmienny

- wartości szarości obrazu są interpretowane jako informacje o wysokości ("góry szarości")
- w miarę sukcesywnego zalewania góra pomiędzy sąsiednimi zbiornikami tworzą się

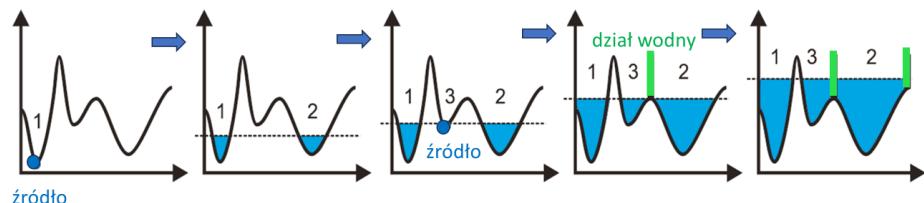


Grafika 14.13: Granice obiektów wyznaczone przez krawędzie obiektów



Grafika 14.14: Negatyw obrazu i jego topografia

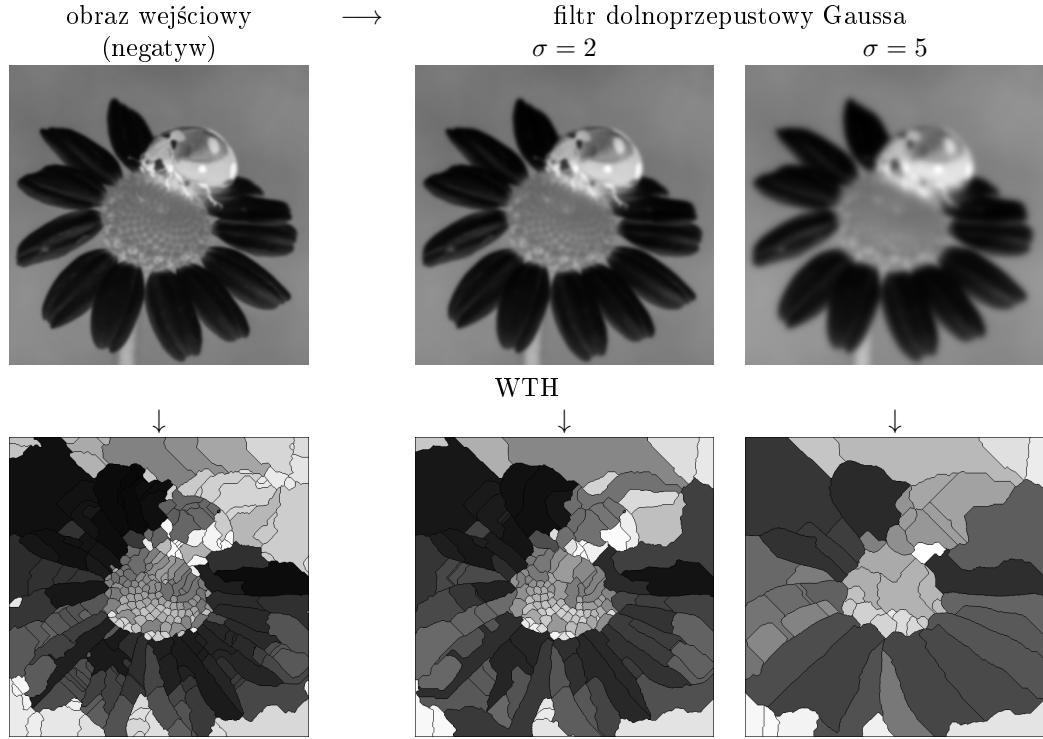
działy wodne



→ zwykle skutkuje to nadmierną segmentacją obrazu, szczególnie w przypadku zaszułonego materiału obrazowego, na przykład danych z tomografii komputerowej

→ obraz należy wstępnie przetworzyć i/lub połączyć segmenty w etapie scalania w oparciu o kryterium podobieństwa

→ grafika 14.15



Grafika 14.15: Transformacja wododziałowa na negatywie obrazu

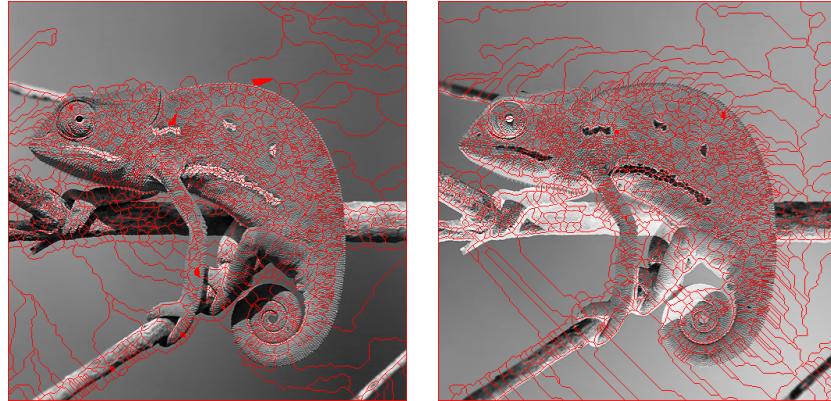
### Algorytm segmentacji wododziałowej (szkic)

Każdy nowo zalany piksel  $(m_w, n_w)$  na wysokości  $h < h_{\text{aktualna}}$  jest

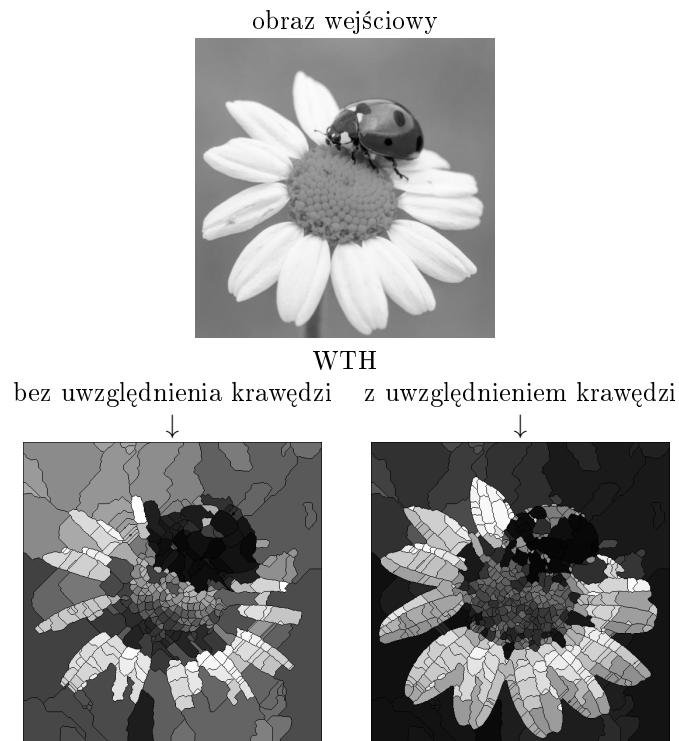
- w izolacji:  
nie sąsiaduje z innymi zalonymi pikselami o wysokości  $h < h_{\text{aktualna}}$   
→ załączek nowego segmentu
- rozszerzeniem:  
przylega do innych zalanych pikseli o wysokości  $h < h_{\text{aktualna}}$  o tej samej etykiecie  
→ piksel jest przypisany do segmentu oznaczonego tą etykietą
- działem wodnym:  
przylega do zalanych pikseli co najmniej dwóch regionów  
→ pikselowi przypisuje się etykietę "dział wodny"  
→ w segmentacji wododziałowej każde lokalne minimum jest źródłem  
→ grafika 14.16

### Wariant WTH z uwzględnieniem krawędzi obiektów

Zlewnie zostają zbudowane wzduż mocnych krawędzi obliczonych np. z gradientu obrazu oryginalnego. → grafika 14.17



Grafika 14.16: Segmentacja wododziałowa obrazu i jego negatywu. Obrazy przetworzone wstępnie filtrem Gaussa ( $\sigma = 2$ )



Grafika 14.17: Transformacja wododziałowa - zlewnie wyznaczone z uwzględnieniem krawędzi obiektów (obrazy przetworzone wstępnie filtrem dolnoprzepustowym Gaussa  $\sigma = 2$ )

### 14.3 Metody obszarowe

- informacja wykorzystywana w czasie segmentacji to jednorodność wewnątrz obiektu
- grafika 14.18



Grafika 14.18: Jednorodne regiony w obrazie

- warunek jednorodności ocenia się podczas segmentacji
- jednorodność jest definiowana w odniesieniu do atrybutów segmentu
- globalne związki można znaleźć dzięki strategii wieloskalowej

#### 14.3.1 Strategie wieloskalowe

**Cel:** analiza obrazu w różnych rozmiarach (poziomach rozdzielczości)

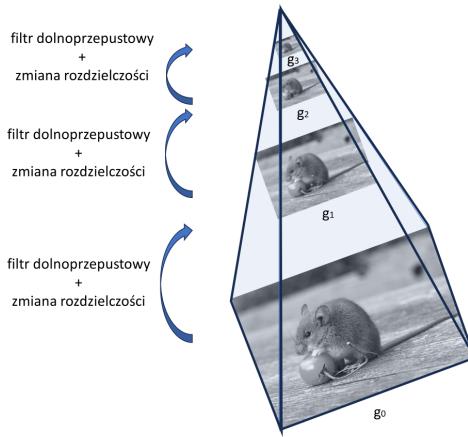
- nieskomplikowane badanie grubych struktur w obrazie
- dokładniejsze badanie drobnych struktur

**Pomysł:** przestrzeń wieloskalowa (podobna do ludzkiej percepcji)

- zawiera oryginalny obraz  $g_0$  i mniej szczegółowe wersje  $g_k$
- generowanie obrazów wieloskalowych (piramid)  $g_0, g_1, g_2, \dots$  poprzez wielokrotne wykonywanie sekwencji kroków:

1. wygładzanie obrazu (za pomocą filtra dolnoprzepustowego, np. filtra Gaussa)
2. **reduce:** kompresja poprzez niższą częstotliwość próbkowania, np. usunięcie co drugiego wiersza i kolumny
3. **expand:** inwersja poprzez interpolację (niebezstratną) w celu porównania obrazów o różnych rozdzielczościach

→ grafika 14.19



Grafika 14.19: Piramida obrazów

### Piramida Gaussa

→ technika stosowana zwłaszcza w syntezie tekstur

Operacja "reduce":

- obraz wejściowy jest skalowany w sposób ciągły za pomocą operacji `reduce`
- każdy piksel kolejnego, wyższego poziomu skalowania reprezentuje 4 piksele bieżącego poziomu
- przed operacją zakres częstotliwości jest redukowany poprzez filtrowanie dolno-przepustowe (Gauss, filtr o współczynnikach dwumianu Newtona)

→ grafika 14.20

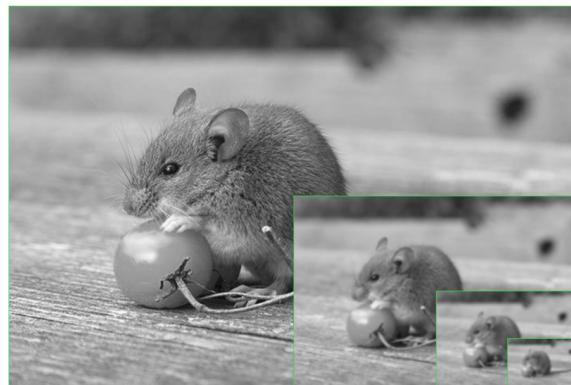
Operacja "expand":

- aby utworzyć poprzedni poziom skalowania, definiuje się operację `expand`
- piksele nowego poziomu skalowania tworzone są poprzez interpolację
- operacja "expand" nie jest bezstratna

→ grafika 14.21

**Przykład:** Segmentacja wododziałowa obrazów piramidy Gaussa

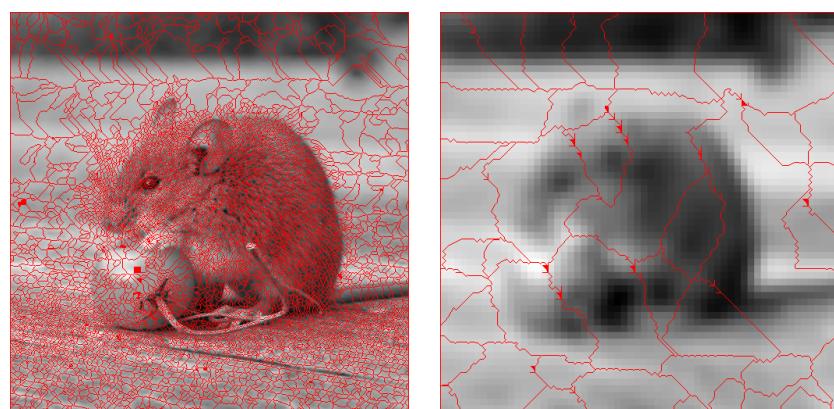
→ grafika 14.22



Grafika 14.20: Piramida Gaussa - etap **reduce**



Grafika 14.21: Piramida Gaussa - etap **expand**



Grafika 14.22: Segmentacja wododziałowa obrazów  $g_0$  i  $g_3$  piramidy Gaussa

### Piramida Laplace'a

→ technika stosowana zwłaszcza w kompresji obrazu

- każdy poziom skalowania  $k$  zawiera tylko różnicę  $g_k - \text{expand}(\text{reduce}(g_k))$
- najniższy poziom nie jest obrazem różnicowym, aby umożliwić rekonstrukcję obrazu o wysokiej rozdzielcości przy użyciu obrazów różnicowych na wyższych poziomach
- reprezentacja bez redundancji

→ grafika 14.23



Grafika 14.23: Piramida Laplace'a

#### 14.3.2 Rozrost obszarów (*Region Growing*)

##### Rozrost prosty

- każdy piksel jest uważany za węzeł w grafie
- sąsiednie punkty, których różnica w kryterium podobieństwa jest mniejsza od progu, są połączone krawędzią
- segmenty obrazu to maksymalnie połączone elementy grafu

Przykład → grafika 14.24

Zalety:

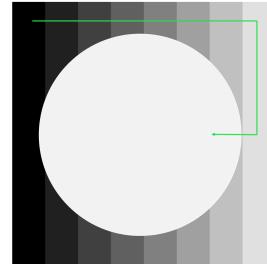
- prosta, szybka, lokalna metoda

Wady:



Grafika 14.24: Obraz i jego graf rozrozstu prostego dla kryterium podobieństwa: max różnica szarości = 5.

- metoda często prowadzi to do błędów w łączeniu segmentów  
→ regiony o różnych kolorach można połączyć łańcuchem wolno łączących się kolorów: wystarczy jedna krawędź w grafie łącząca jeden region z sąsiednim;



- z powodu nadmiernych fuzji metoda nie jest praktyczna w przypadku złożonych scen

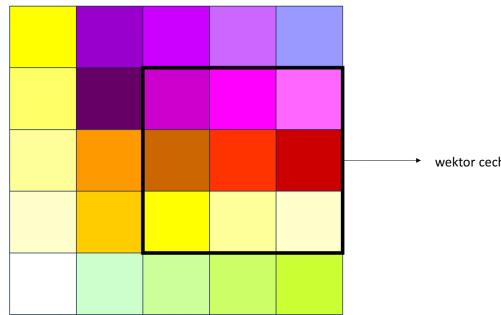
### Rozrost hybrydowy

→ grafika 14.25

- decyzja o tym, czy dwa piksele łączy krawędź w grafie, podejmowana jest na podstawie wartości kolorów sąsiedztwa pikseli
- każdemu pikselowi przypisany jest wektor charakterystyczny, który zależy od jego sąsiedztwa  $k \times k$
- piksele są "podobne", jeśli ich sąsiedztwo jest "podobne"

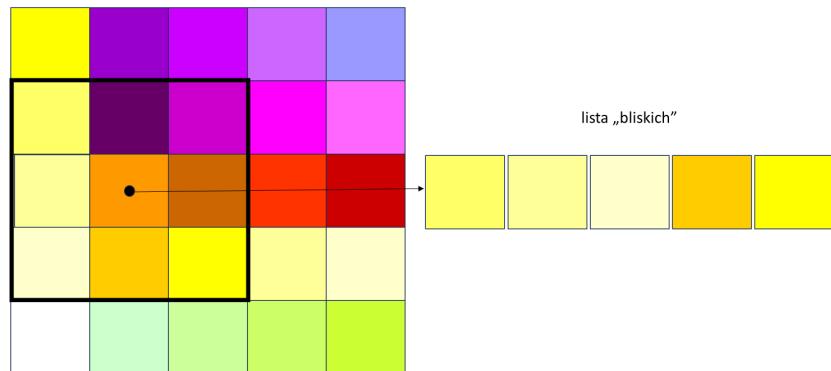
**Przykład:** metoda Levine i Leemet'a  
→ oparta na idei najbliższego sąsiada

- każdy piksel oblicza listę  $N$  pikseli z sąsiedztwa  $k \times k$ , które są do niego najbardziej podobne → lista  $N$  "bliskich" (grafika 14.26)

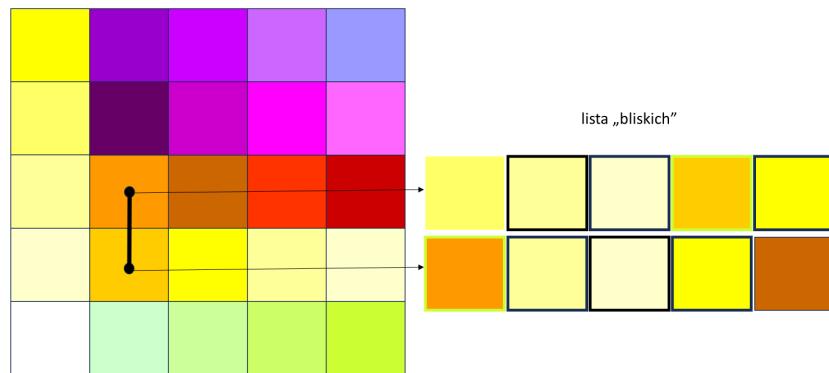


Grafika 14.25: Rozrost hybrydowy - wektor cech na podstawie sąsiedztwa piksela  $3 \times 3$

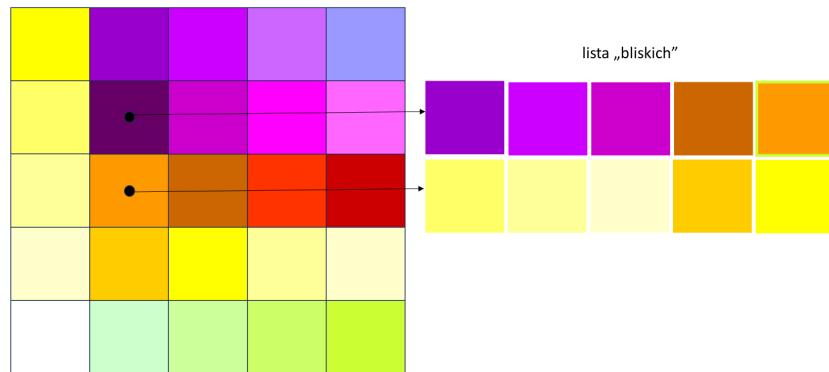
- można zastosować dowolną miarę podobieństwa pikseli dla obrazów kolorowych np. odległość euklidesową
- dwa sąsiednie piksele są połączone krawędzią, jeżeli
  - każdy piksel znajduje się na liście "bliskich" drugiego i
  - jest wystarczająca liczba pikseli, które znajdują się na listach "bliskich" obu pikseli — grafiki 14.27, 14.28



Grafika 14.26: Lista  $N = 5$  "bliskich" piksela w sąsiedztwie  $3 \times 3$



Grafika 14.27: Połączenie krawędzią pikseli - spełnione oba warunki połączenia hybrydowego.

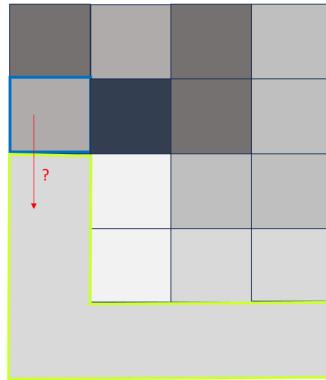


Grafika 14.28: Brak połączenie krawędzią pikseli

### Rozrost centroidalny

→ grafika 14.29

- piksel jest porównywany ze średnią istniejącego, ale niekoniecznie ukończonego, sąsiadującego segmentu
- jeżeli wartość piksela i średnia wartość segmentu są wystarczająco podobne, piksel jest dodawany do segmentu, a wartość średnia jest aktualizowana



Grafika 14.29: Łączenie centroidalne

---

#### **Algorytm Region Growing (szkic)**

1. podzielenie obrazu na komórki inicjalne ( $1 \times 1$ ,  $2 \times 2$ ,  $4 \times 4$ )
2. obliczenie statystycznej miary intensywności we wszystkich komórkach (np. średniej wartości szarości)
- poczynając od lewego górnego rogu:
3. porównanie statystyk komórki z każdą z sąsiednich komórek, aby zdecydować, czy są one podobne,  
jeśli porównanie się powiedzie:  
  połączenie komórki z segmentem i aktualizacja miary statystycznej dla segmentu
4. segment rośnie poprzez porównywanie ze wszystkimi swoimi sąsiadami, aż do momentu, gdy nie będzie już można dodać więcej sąsiadów  
  → segment jest oznaczony jako gotowy region
5. następna nieprzetworzona komórka jest przetwarzana w krokach 3 – 4, aż do przetworzenia wszystkich komórek

---

Podstawowa sekwencja algorytmu *Region Growing* jest taka sama we wszystkich jego wariantach. Udoskonalone są kryteria podobieństwa stosowane przy podejmowaniu decyzji o łączeniu regionu i piksela (lub komórki).

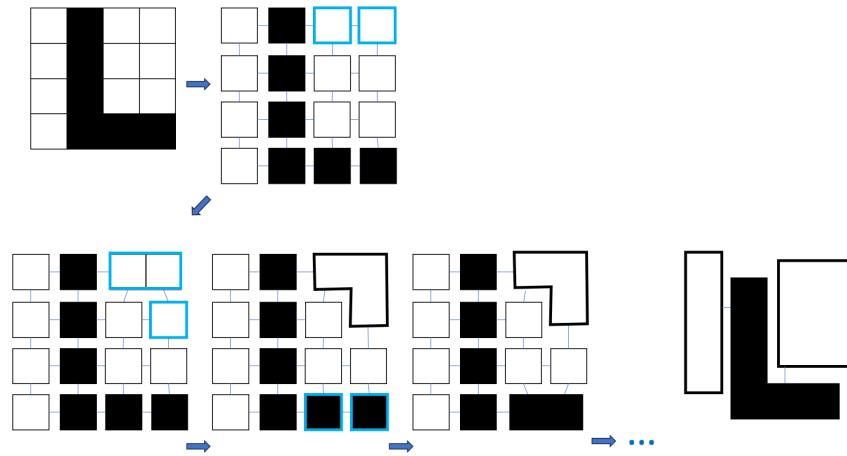
Wnioski:

- kolejność przetwarzania pikseli wpływa na wynik segmentacji: dwa różne punkty początkowe dają różne wyniki segmentacji

- kluczowe znaczenie dla wyniku segmentacji ma wybór kryteriów łączenia oraz wybór prógów

### 14.3.3 Łączenie obszarów (*Region Merging*)

→ grafika 14.30



Grafika 14.30: Łączenie obszarów

- inicjalnie każdy piksel jest deklarowany jako segment
- dwa sąsiednie regiony są łączone, jeśli razem spełniają kryterium podobieństwa
  - różnica wartości szarości w regionie
  - prawdopodobieństwo, że oba regiony mają takie same rozkłady wartości szarości
  - ...
- segmentacja jest zakończona, gdy nie ma już dwóch regionów, które można połączyć
- wyniki pośrednie są przechowywane w grafie sąsiedztwa regionu (*Region Adjacency Graph*, RAG)

**Przykład: Region Merging i strategia wieloskalowa → strategia hybrydowa**

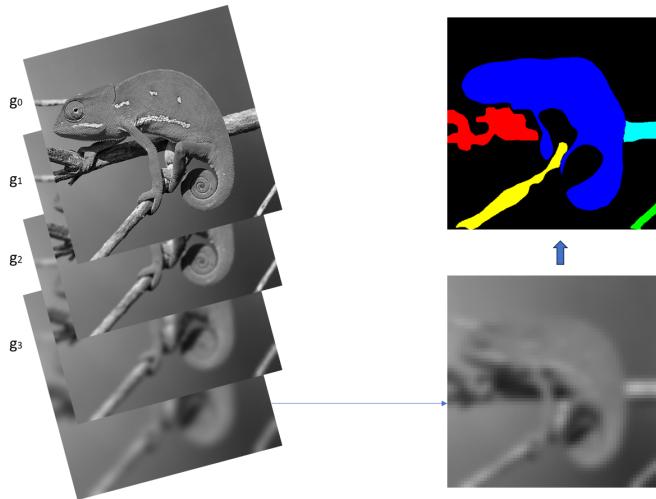
Założenie modelu:

znany jest najgrubszы poziom skali, na którym ujawniają się właściwości istotne dla segmentacji.

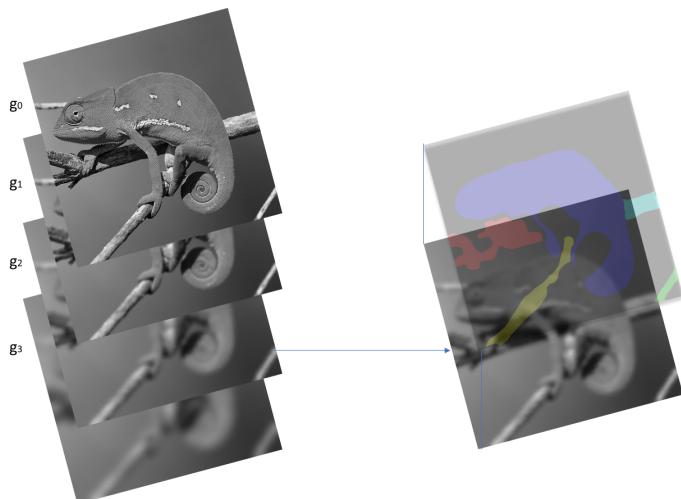
Proces segmentacji:

- łączenie regionów (Region Merging) w skali zgrubnej → grafika 14.31

- przeniesienie wyniku segmentacji na wyższy poziom skali  $\rightarrow$  grafika 14.32
- wszystkie piksele sąsiadujące z pikselami innego segmentu są ponownie sprawdzane
- proces kończy się po osiągnięciu najwyższego poziomu skalowania



Grafika 14.31: Łączenie regionów (Region Merging) w skali zgrubnej



Grafika 14.32: Przeniesienie wyniku łączenia regionów na wyższy poziom skali

#### 14.3.4 Podział i łączenie (*Split & Merge*)

→ Algorytm Horowitza i Pavlidisa  
 → grafika 14.33

Warunek startowy: cały obraz jest jednym segmentem

- segment dzieli się na 4 podsegmenty, o ile nie spełnia określonego kryterium jednorodności
- segmenty sąsiadujące ze sobą łączy się, jeśli po połączeniu nadal spełniają kryterium jednorodności
- rezultatem segmentacji jest kompletny rozkład obrazu bez nakładania się segmentów (segmentacja zgodnie z definicją)  
     → rozkład kończy się najpóźniej na poziomie piksela

#### Algorytm *Split & Merge* (szkic)

Niech  $H$  będzie kryterium jednorodności.

1. obraz jest dzielony na  $2^n$  kwadratowe obszary o jednakowej wielkości
2. *split*: jeżeli dla obszaru  $R$  w tej strukturze  $H(R) = FALSE$ , region jest dzielony na cztery podregiony
3. *merge*: jeżeli dla czterech regionów kwadrantu  $R_{k_1}, R_{k_2}, R_{k_3}, R_{k_4}$ :  $H(R_{k_1} \cup R_{k_2} \cup R_{k_3} \cup R_{k_4}) = TRUE$ , regiony są łączone w jeden
4. *grouping*: jeżeli dla dwóch sąsiadujących regionów  $R_i$  i  $R_j$  (regiony niekoniecznie tej samej wielkości)  $H(R_i \cup R_j) = TRUE$ , regiony zostają połączone

#### Kryteria jednorodności

- obrazy w skali szarości
  - różnica między maksymalną a minimalną wartością szarości w każdym z 4 regionów poniżej danego progu  $T$  (Horowity i Pavlidis)
  - ...
- kryteria dla fazy *split* i *merge* mogą być różne
- obrazy RGB
  - podział i łączenie na podstawie wariancji kanałów (Dubuisson i Jain):

- \* dla każdego kwadratowego regionu  $R$  o wymiarach  $z \times z$  obliczane są wartości średnie  $m_I$  i wariancje  $v_I$ ,  $I \in \{R, G, B\}$

$$\begin{aligned} m_I &= \frac{1}{z^2} \sum_{(m,n) \in R} I(m,n), \quad I \in \{R, G, B\} \\ v_I &= \frac{1}{z^2} \sum_{(m,n) \in R} (I(m,n) - m_I)^2, \quad I \in \{R, G, B\} \end{aligned}$$

- \* regiony, które nie są jednorodne, tzn. ich wariancja jest zbyt duża ( $v_G + v_R + v_B \geq T_v$ ) są dzielone (*split*)
- \* 4 sąsiadujące regiony, które są jednorodne tzn.

$$|m_{R_i} - m_{R_j}| + |m_{G_i} - m_{G_j}| + |m_{B_i} - m_{B_j}| \leq T_m \text{ dla wszystkich } i, j$$

zostają połączone

- \* sąsiadujące regiony o wartościach średnich kanałów  $m_I, M_I$ ,  $I \in \{R, G, B\}$ , dla których

$$|m_R - M_R| + |m_G - M_G| + |m_B - M_B| \leq T_g$$

zostają połączone (*grouping*)

- \* progi  $T_v, T_m, T_g$  są wyznaczane empirycznie (standard  $T_v = 25, T_m = 25, T_g = 16$ )

— ...

### **Struktura danych: drzewo czwórkowe (*quadtree*)**

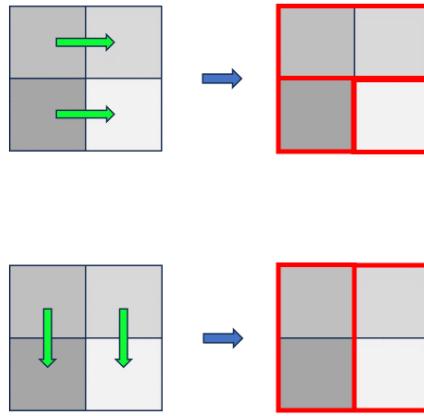
→ grafika 14.34

→ wartość charakterystyki jednorodności regionu jest przechowywana w odpowiednim liściu drzewa quad

### **Problemy (*Merge, Split & Merge*)**

- etap łączenia nie zawsze jest jednoznaczny w przypadku cechy wzgórnej jednorodności

- nie zawsze udaje się znaleźć minimalną liczbę segmentów



#### 14.3.5 Klastrowanie

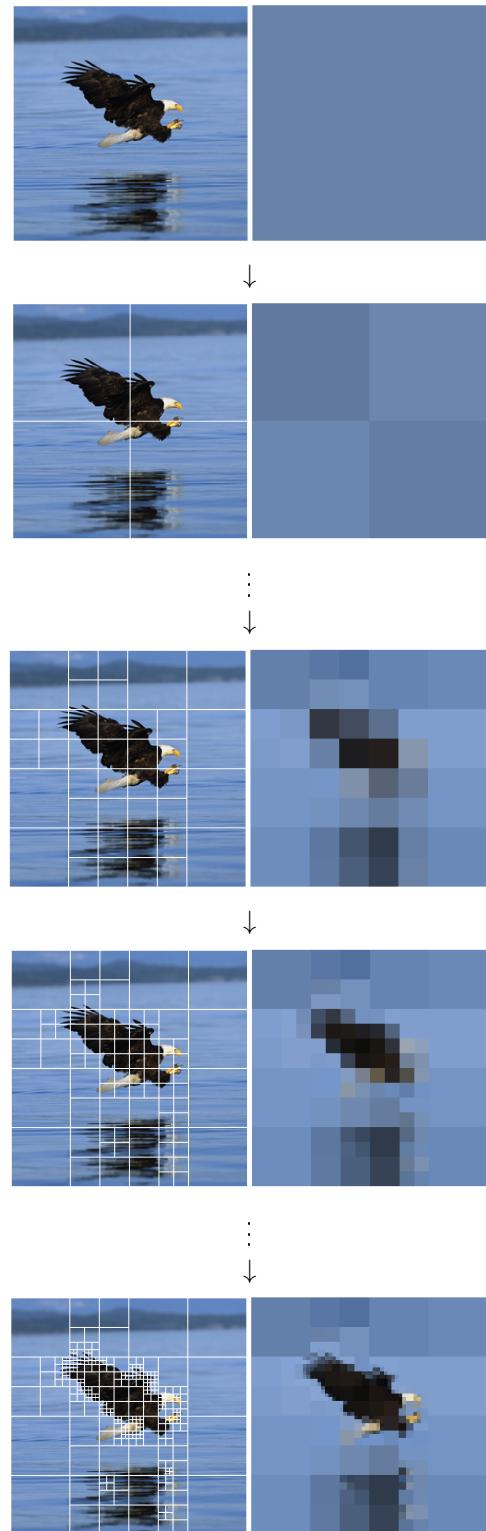
→ podział obrazu na  $k$  klastry, grafika 14.35

##### Algorytm klastrowania (szkic)

1. wybór  $k$  środków klastrów (losowo lub w oparciu o metodę heurystyczną)
2. każdy piksel na obrazie jest przypisany do klastra, który minimalizuje różnicę między pikselem a środkiem klastra (kolor, intensywność, lokalizacja piksela lub ważona kombinacja tych charakterystyk)
3. ponowne obliczenie nowych środków klastrów poprzez uśrednienie wszystkich pikseli w klastrze
4. powtarzanie kroków 2 i 3 aż do terminacji algorytmu (żadne piksele nie zmieniają klastra)

→ grafika 14.36

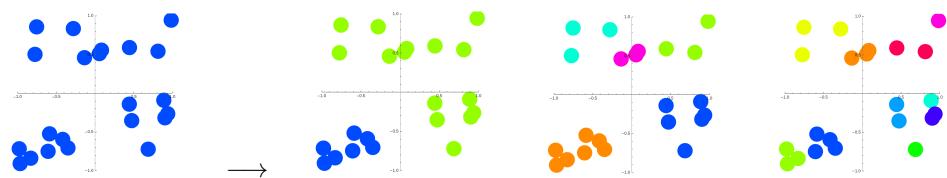
→ powszechnie stosuje się algorytm wraz ze strategią wieloskalową (zmniejszenie rozdzielcości obrazu, obliczenie klastrów, a następnie przeskalowanie wzwyż).

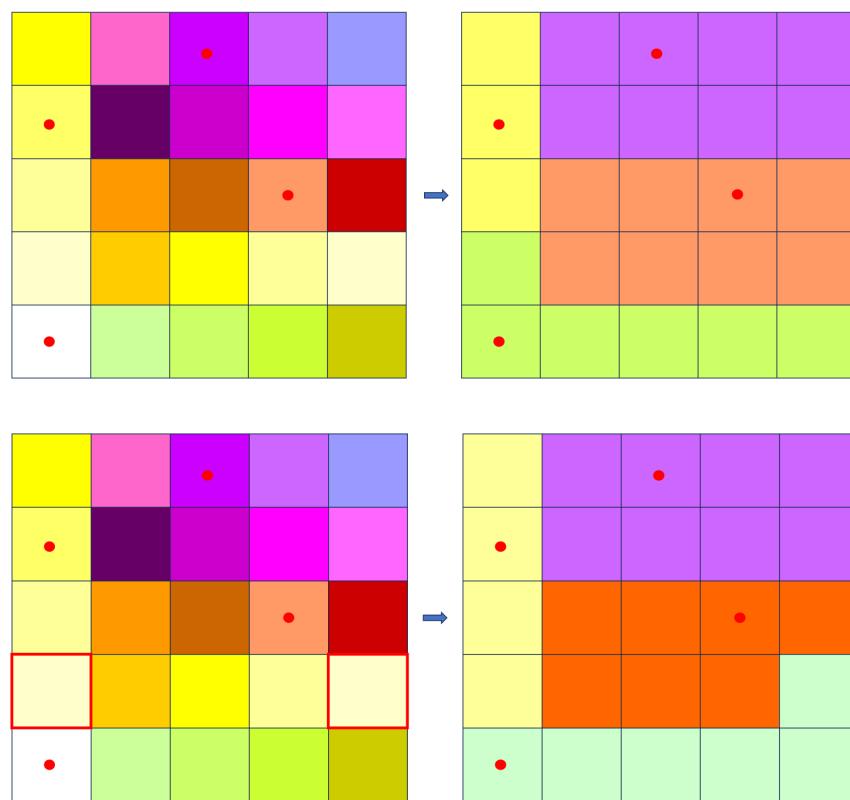


Grafika 14.33: Algorytm segmentacji dziel i łącz.  
Źródło: <https://demonstrations.wolfram.com/QuadTreeImageDecomposition/>



Grafika 14.34: Drzewo czwórkowe

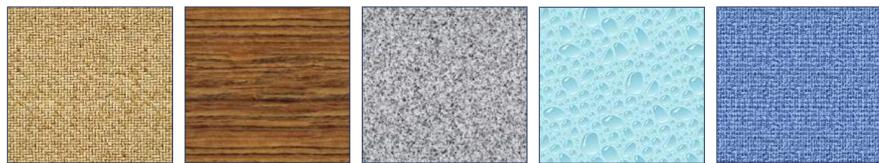
Grafika 14.35: Podział punktów w przestrzeni  $2D$  na  $k = 2, 5, 10$  klastrów według funkcji podobieństwa (odległość euklidesowa)



Grafika 14.36: Algorytm klastrowania

## 14.4 Tekstura

→ grafika 14.37



Grafika 14.37: Tekstura powierzchni

- Tekstura – wzór na powierzchni
  - nie ma definicji tekstury
  - istnieje wiele miar dla tekstury
  - tekstura jest z natury zależna od skali
  - tekstura jest właściwością grupy pikseli
- Miary dla tekstury
  - strukturalne (skład elementów tekstury – texel)
  - stochastyczne (charakteryka rozkładu wartości szarości)
  - widmowe (charakterystyczne atrybuty w dziedzinie częstotliwościowe)

Niektóre obiekty obrazu nie mają jednolitego koloru, ale raczej jednolitą teksturę → grafika 14.38.

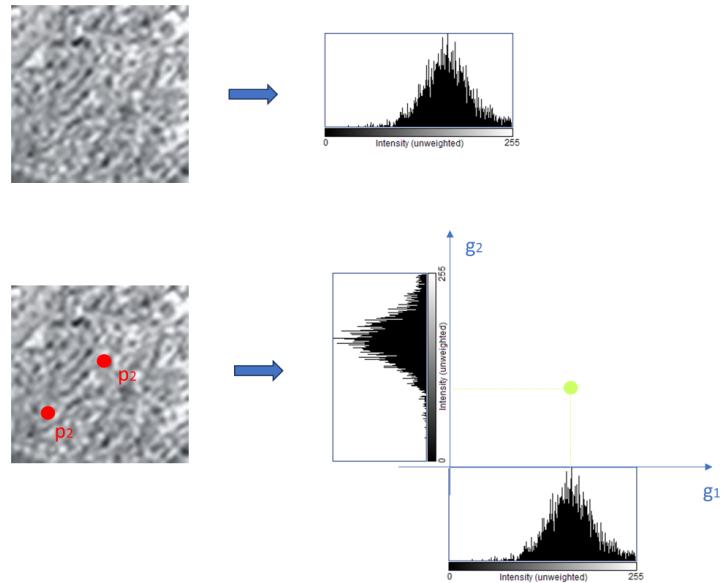
Aby obiekty te nie zostały podzielone na wiele małych obiektów w oparciu o teksturę, stosuje się metody mające na celu rozwiązanie tego problemu.



Grafika 14.38: Tekstura na obrazie

### Macierz szarości (Co-Occurrence-Matrix, gray level co-occurrence matrix, GLCM)

- macierz  $\mathbf{W}_\rho$  kombinacji wartości szarości, dwuwymiarowy histogram
- grafika 14.39

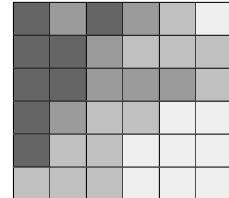


Grafika 14.39: Konstrukcja dwuwymiarowego histogramu

- $\rho$ : relacja między pikselami  $p_1 = (m_1, n_1)$  i  $p_2 = (m_2, n_2)$

Przykład:

$$\rho(x, y) = (x + 1, y) \quad g :$$



- $a_{g_1 g_2}$ : częstość kombinacji wartości szarości  $(g_1, g_2) = (g(m_1, n_1), g(m_2, n_2))$  w relacji  $\rho$

Przykład:

$$\mathcal{G} \in \{0, 1, 2, 3\}$$

$g :$

$$\rho(x, y) = (x + 1, y)$$

■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■

0	1	0	1	2	3
0	0	1	2	2	2
0	0	1	1	1	2
0	1	2	2	3	3
0	2	2	3	3	3
2	2	2	3	3	3

$$\Rightarrow a_{1,2} = 4$$

0	1	0	1	2	3
0	0	1	2	2	2
0	0	1	1	1	2
0	1	2	2	3	3
0	2	2	3	3	3
2	2	2	3	3	3

$$\Rightarrow a_{1,0} = 1$$

0	1	0	1	2	3
0	0	1	2	2	2
0	0	1	1	1	2
0	1	2	2	3	3
0	2	2	3	3	3
2	2	2	3	3	3

- macierz  $\mathbf{W}_\rho = [a_{g_1 g_2}]$

Przykład:

$g :$

$$\rho(x, y) = (x + 1, y)$$

■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■

0	1	0	1	2	3
0	0	1	2	2	2
0	0	1	1	1	2
0	1	2	2	3	3
0	2	2	3	3	3
2	2	2	3	3	3

$$\mathbf{W}_\rho = \begin{bmatrix} 2 & 5 & 1 & 0 \\ 1 & 2 & 4 & 0 \\ 0 & 0 & 6 & 4 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

- (zwyczajowa) reprezentacja macierzy  $\mathbf{W}_\rho$  jako obraz o wymiarach  $\mathcal{G} \times \mathcal{G}$

$$\mathbf{W}_\rho = \begin{bmatrix} 2 & 5 & 1 & 0 \\ 1 & 2 & 4 & 0 \\ 0 & 0 & 6 & 4 \\ 0 & 0 & 0 & 5 \end{bmatrix} \rightarrow \begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array}$$

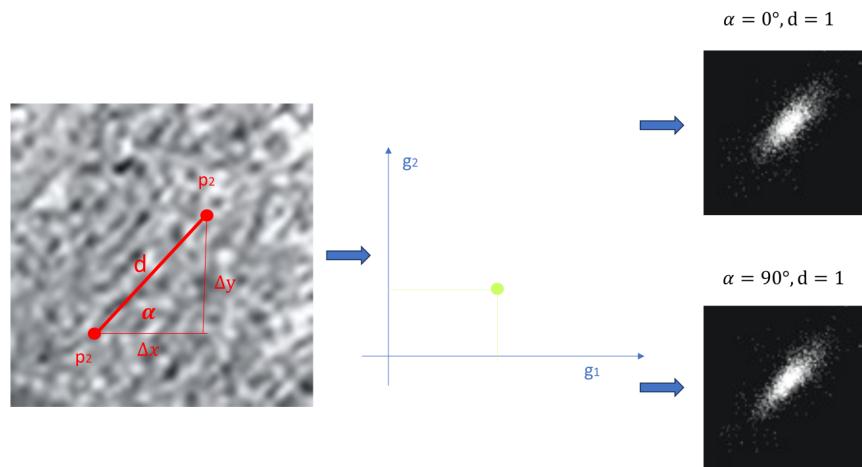
- w obrazie o strukturze powierzchni o wysokim kontraście w lewym dolnym i prawym górnym rogu znajdują się wysokie wartości współczynników macierzy  $\mathbf{W}_\rho$ , natomiast obraz o dużych, monotonnych obszarach ma silną przekątną główną

#### 14.4.1 Macierz szarości i miary tekstuury Haralicka (*Co-Occurrence-Matrix, Haralick features*)

→ macierz  $\mathbf{W}_\rho$  kombinacji wartości szarości, dwuwymiarowy histogram

- relacja  $\rho$  między pikselami:
  - $\rho(x, y) = (x + \Delta x, y + \Delta y)$
  - $\rho(x, y) = (x + d \cos \alpha, y + d \sin \alpha)$
- większość pikseli nie jest skorelowana na dużych odległościach, dlatego zazwyczaj wybór wartości  $d = 1, d = 2$
- jeżeli podejrzewa się korelację między pikselami o większej odległości, należy zastosować strategię wieloskalową

→ grafiki 14.40, 14.41



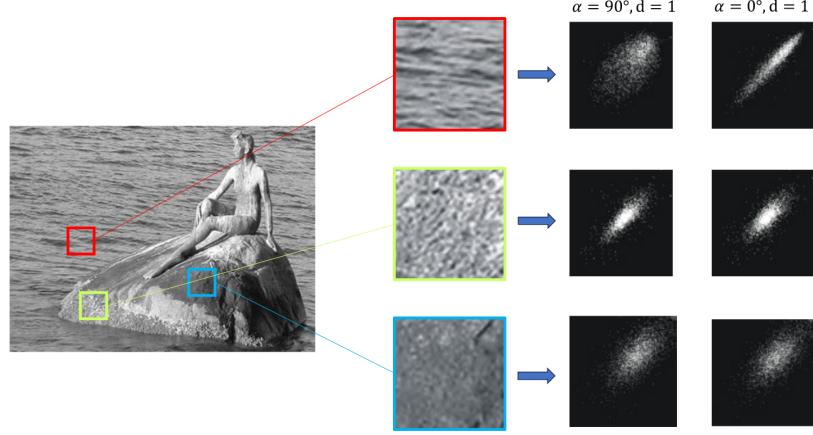
Grafika 14.40: Co-Occurrence-Matrix z relacją  $\rho(x, y) = (x + d \cos \alpha, y + d \sin \alpha)$  dla  $\alpha = 0^\circ, 90^\circ$  i  $d = 1$

#### Miary tekstuury Haralicka (*Haralick features*)

Miary (cechy) tekstuury Haralicka można obliczyć z macierzy szarości. Są one znaczącymi parametrami dla różnych tekstur.

- energia

$$E = \sum_{g_1=0}^{G-1} \sum_{g_2=0}^{G-1} \mathbf{W}_\rho^2(g_1, g_2)$$



Grafika 14.41: Co-Occurrence-Matrix wybranych tekstur na obrazie. Źródło: Butz/Hoppe "Computergrafik 1", LMU

- kontrast

$$K = \sum_{g_1=0}^{G-1} \sum_{g_2=0}^{G-1} (g_1 - g_2)^2 \cdot \mathbf{W}_\rho(g_1, g_2)$$

- entropia

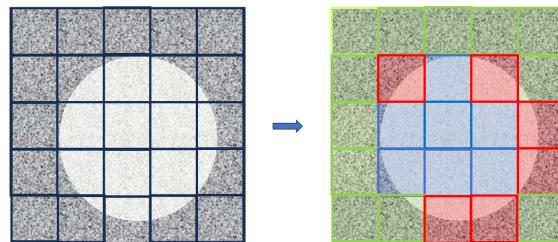
$$S = \sum_{g_1=0}^{G-1} \sum_{g_2=0}^{G-1} \mathbf{W}_\rho(g_1, g_2) \cdot \log(\mathbf{W}_\rho(g_1, g_2))$$

- jednorodność

$$H = \sum_{g_1=0}^{G-1} \sum_{g_2=0}^{G-1} \frac{\mathbf{W}_\rho(g_1, g_2)}{1 + |g_1 - g_2|}$$

### Specyfika cech tekstur

- obliczenie cech tekstury na podstawie dowolnych regionów.
- segmentacja

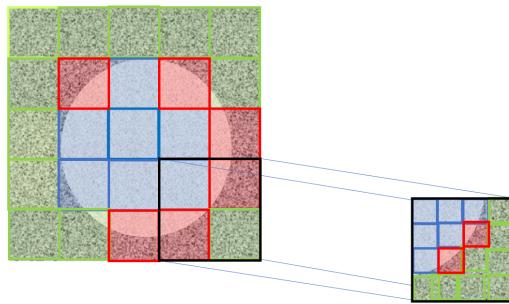


- wyznaczenie niezawodności miar tekstury
- re-segmentacja

### Niezawodność cech tekstur

Persystencja – trwałość:

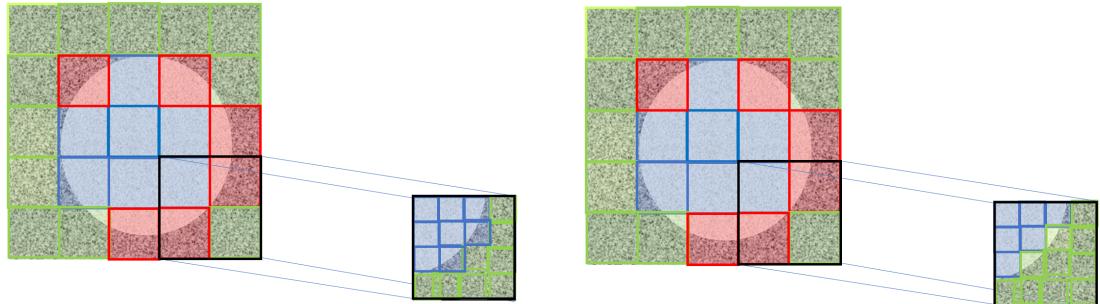
- jeśli region jest rozłożony na części i cecha zostaje obliczona w podobszarach, wówczas cecha tekstury w podobszarze powinna przyjąć te same wartości, co w pierwotnym regionie



- z reguły obliczona miara ma charakter szacunkowy, a jej jakość zależy od wielkości regionu

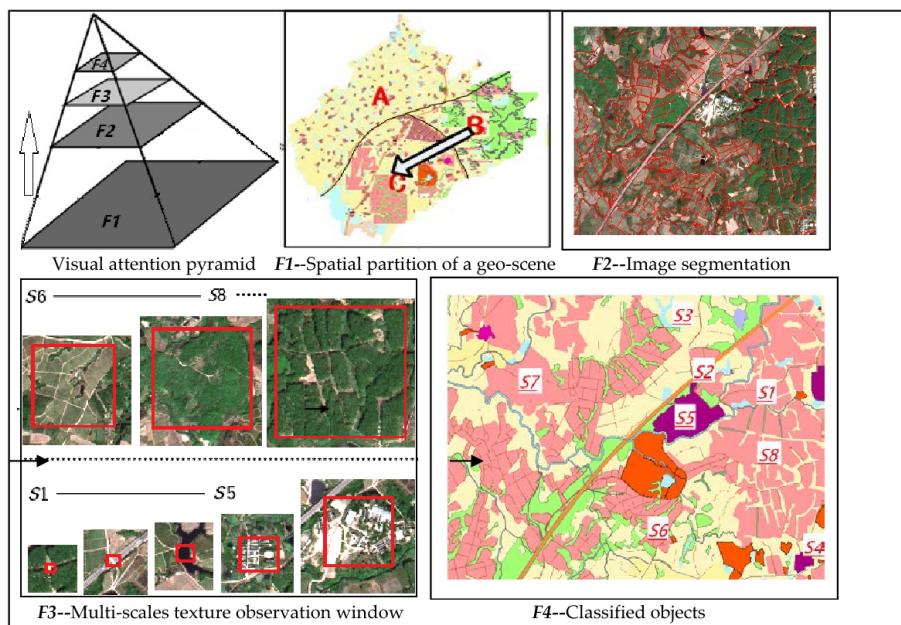
### Strategia: re-segmentacja

- ponowne obliczenie charakterystyki tekstury kompletnych segmentów
- segmenty o dużej zawodności
  - przypisanie do wszystkich sąsiednich niezawodnych segmentów



- wybór najbardziej podobnej kombinacji charakterystyk
- obliczenie charakterystyk można wykonać na wielu poziomach skalowania (tzn. w połączeniu ze strategią wieloskalową)

Przykład: badanie tekstury w geoanalizie obrazu → grafika 14.42



Grafika 14.42: Tekstura powierzchni w geoanalizie obrazu. Źródło: Zeying Lan, Yang Liu, "Study on Multi-Scale Window Determination for GLCM Texture Description in High-Resolution Remote Sensing Image Geo-Analysis Supported by GIS and Domain Knowledge", ISPRS Int. J. Geo Inf.