

Przetwarzanie obrazów 7

- Handout

SL 2024/2025

Wydział Matematyki, Fizyki i Informatyki



Dr.-Ing. A. Rowińska-Schwarzeweller
agnieszka.rowinska-schwarzeweller@ug.edu.pl

23. April 2025

Spis treści

13 Wykrywanie krawędzi, grzbietów i kątów w obrazie	2
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>)	22
13.3 Detektor krawędzi Canny'ego (<i>Canny edge detector</i>)	23
13.4 Detekcja krawędzi w przestrzeni RGB	28
13.5 Detekcja grzbietów (<i>Ridge Detection</i>)	33
13.5.1 Wykrywanie grzbietów z macierzą Hessego	34
13.6 Detekcja kątów obiektów (<i>Corner detection</i>)	40
13.6.1 Harris Corner Detectior	40

13

Wykrywanie krawędzi, grzbietów i kątów w obrazie

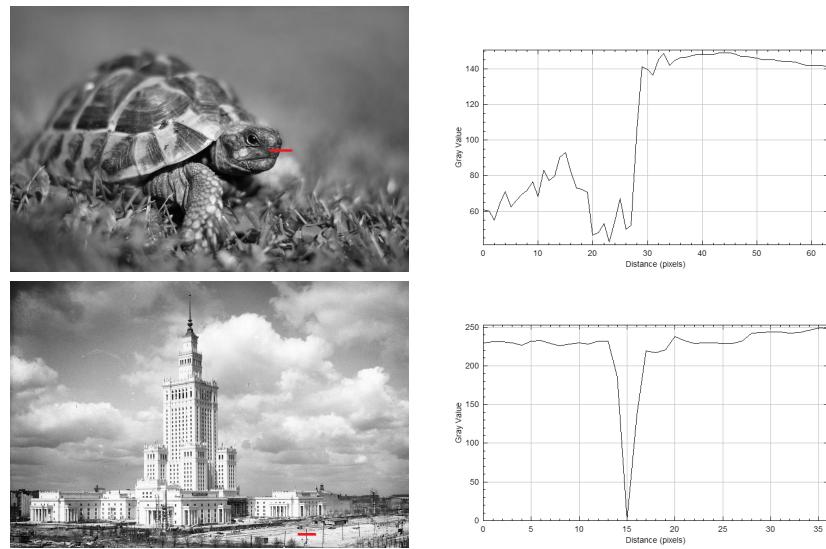
Wykrywanie krawędzi jest podstawowym narzędziem w przetwarzaniu obrazu, widżeniu maszynowym i widzeniu komputerowym, szczególnie w obszarach wykrywania i ekstrakcji cech.

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

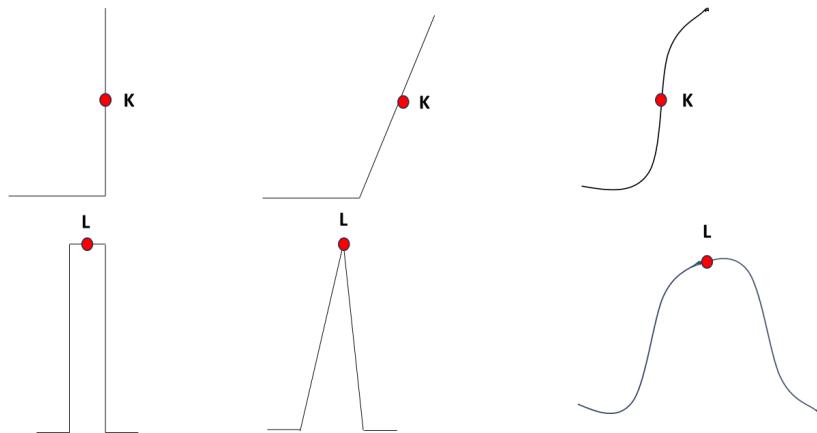


Grafika 13.1: Krawędzie na obrazie.

- 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



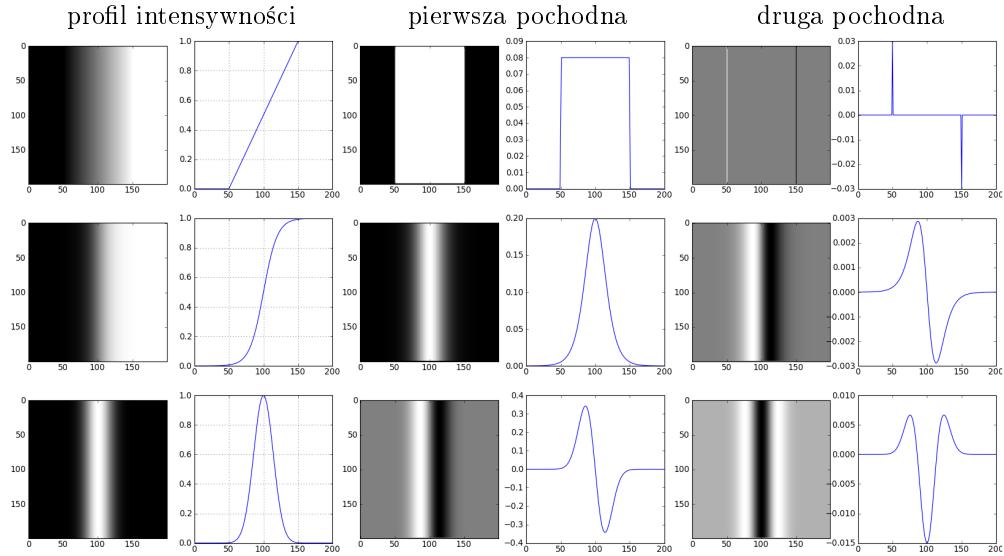
Grafika 13.2: Rzeczywiste krawędzie i linie na obrazie.



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

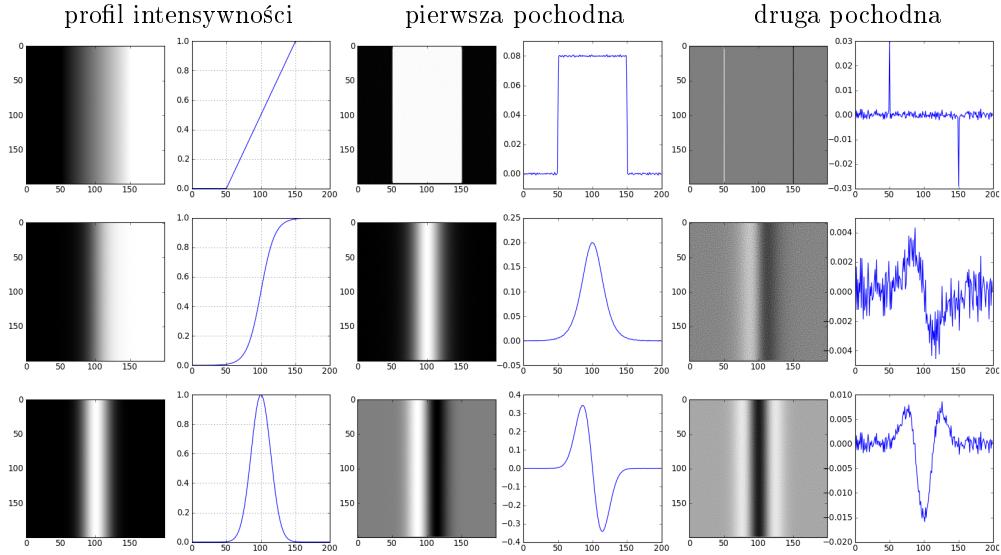
Orientacja krawędzi

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

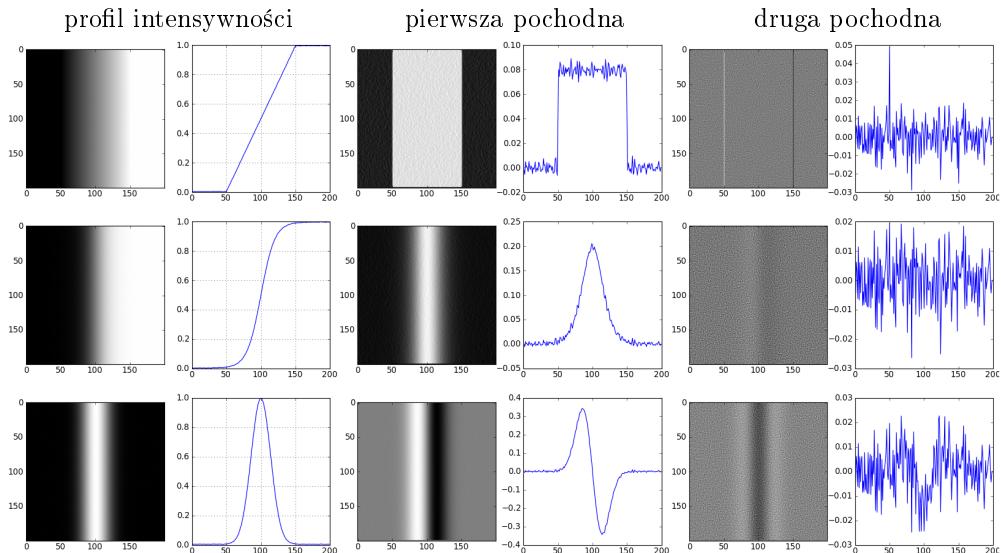
Ogólna procedura ekstrakcji krawędzi:

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

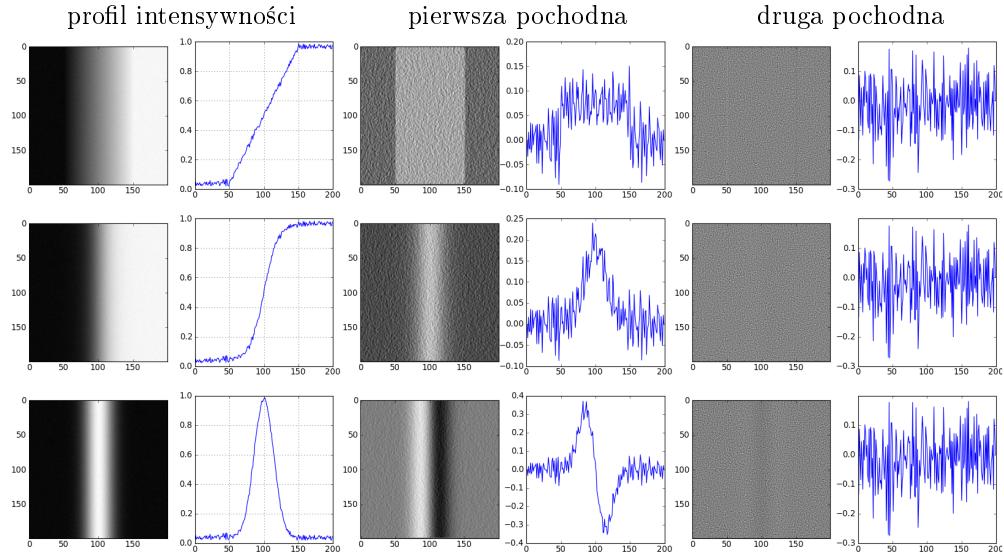
1. wygładzanie (filtr dolnoprzepustowy)
↪ ponieważ oryginalny obraz jest zwykle zaszumiony, bez wygładzenia wykryte byłoby zbyt wiele artefaktów



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

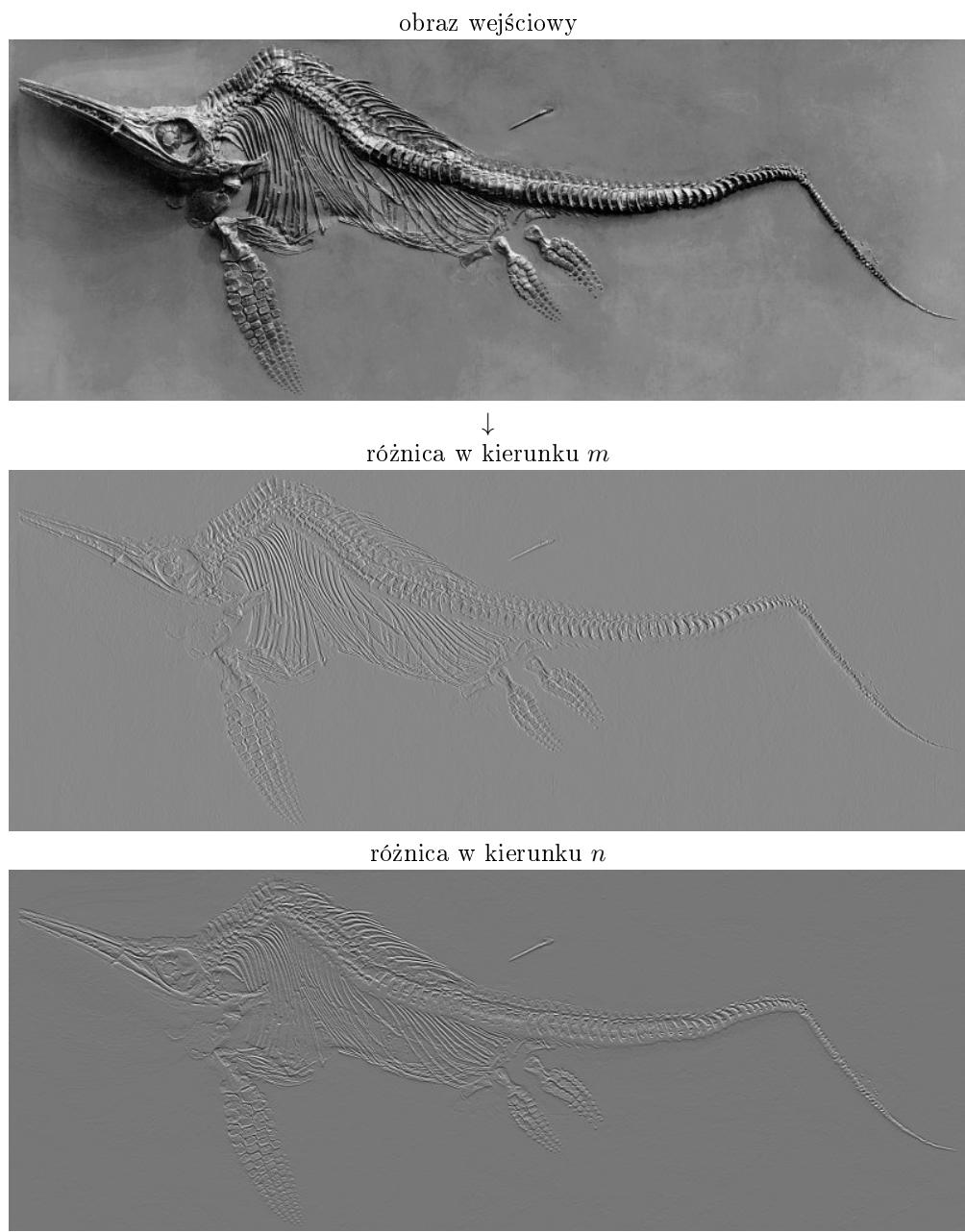
→ filtry liniowe

Operatory do obliczania pierwszej pochodnej:

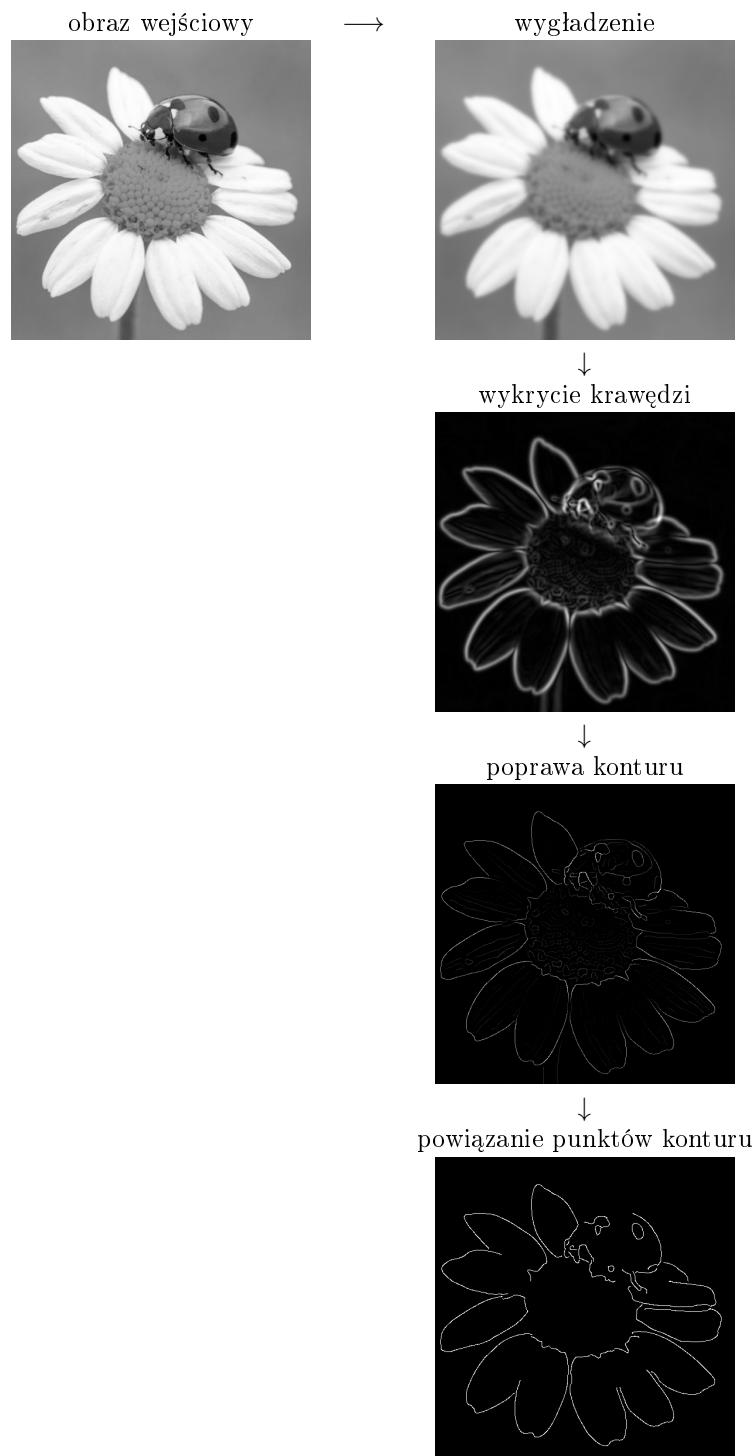
- Sobel
- Prewitt
- Roberts
- Kirsch

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 :

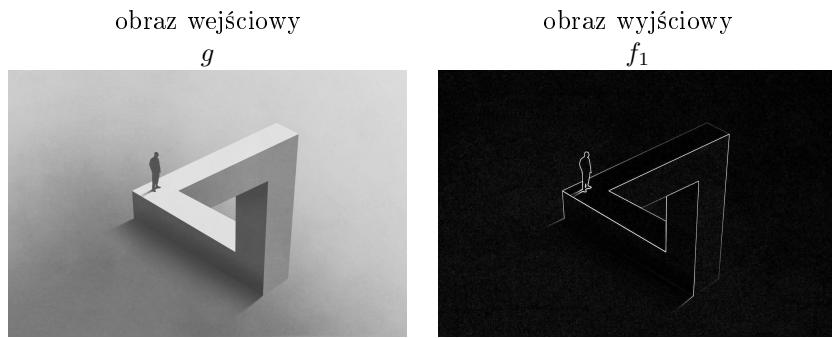
$$h_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad h_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

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

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

lub

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



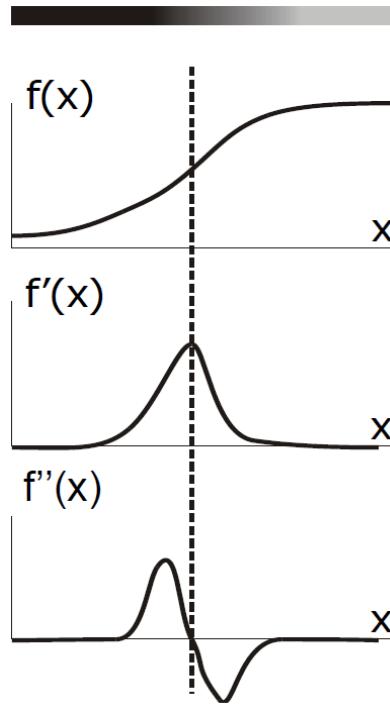
Grafika 13.10: Wykrywanie krawędzi filtrem Sobel'a

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

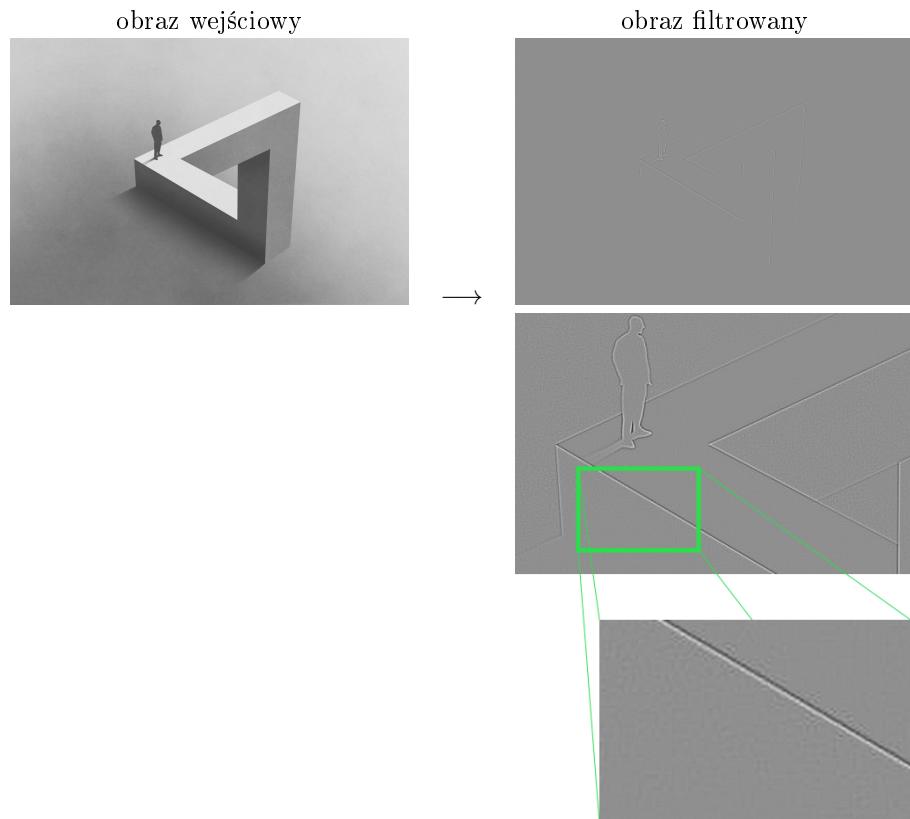
$$h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Wyznaczenie przejść pochodnej przez 0 → grafika 13.11

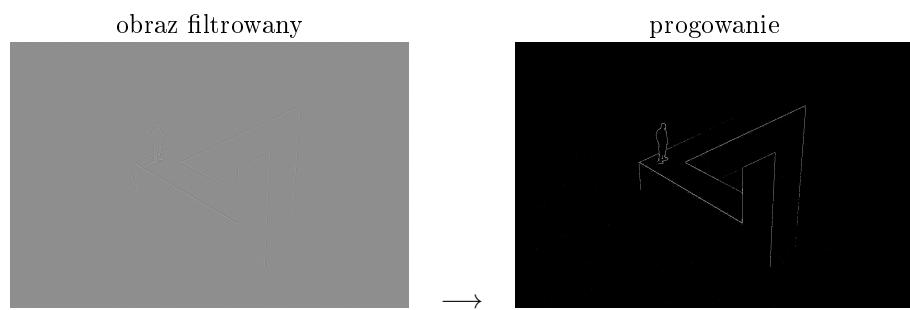
Progowanie obrazu wynikowego → grafika 13.12

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.

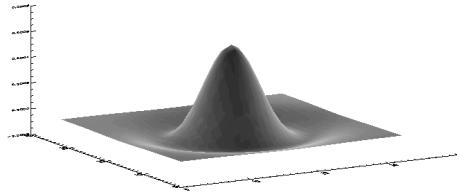
Funkcja Laplace'a funkcji Gaussa (LoG)

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) ∇^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 $k = 3, 5, 7, \dots$

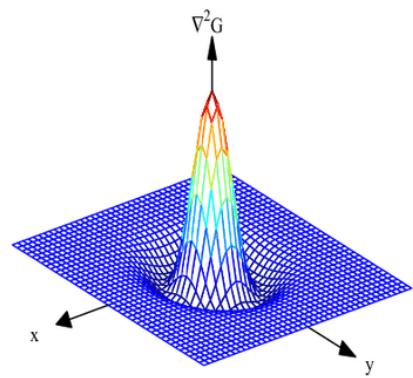
Przykładowe jądro o wielkości 7×7 , tj. dyskretne przybliżenie ciągłego LoG z odchyleniem standardowym $\sigma = 1.6$ (jądro znormalizowane do 1, a następnie pomnożone przez 255):

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

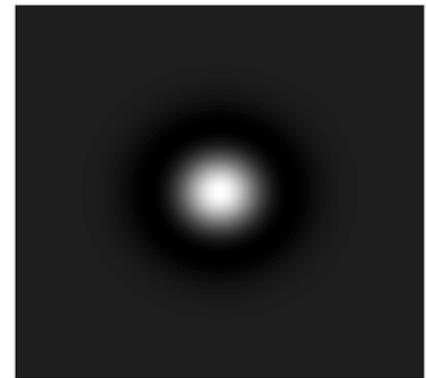
Przykład: → grafika 13.1.3

Zalety operatora LoG

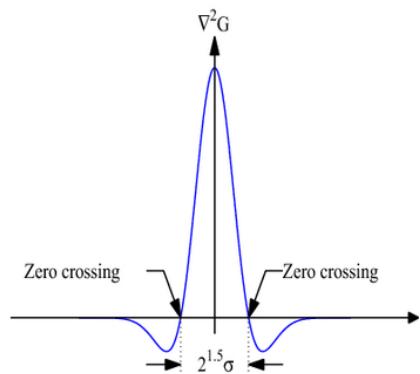
- wygładzanie i podkreślanie krawędzi w jednym operatorze
- mniejsza wrażliwość na szum dzięki wygładzaniu
- funkcja Gaussa nie generuje efektu Gibbsa
- niezmiennik rotacji
- separowalny



(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) 5×5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

Grafika 13.13: Operator LoG 5×5 , $\sigma = 1.5$. 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 3×3

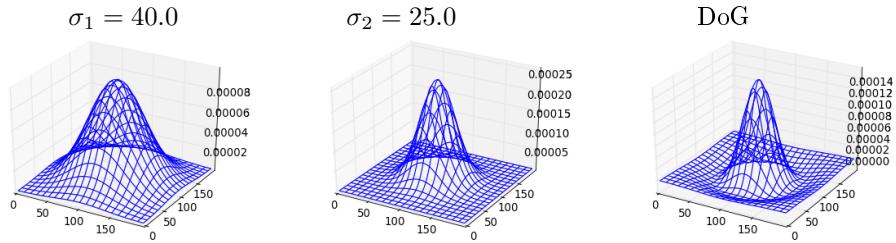
$$h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

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

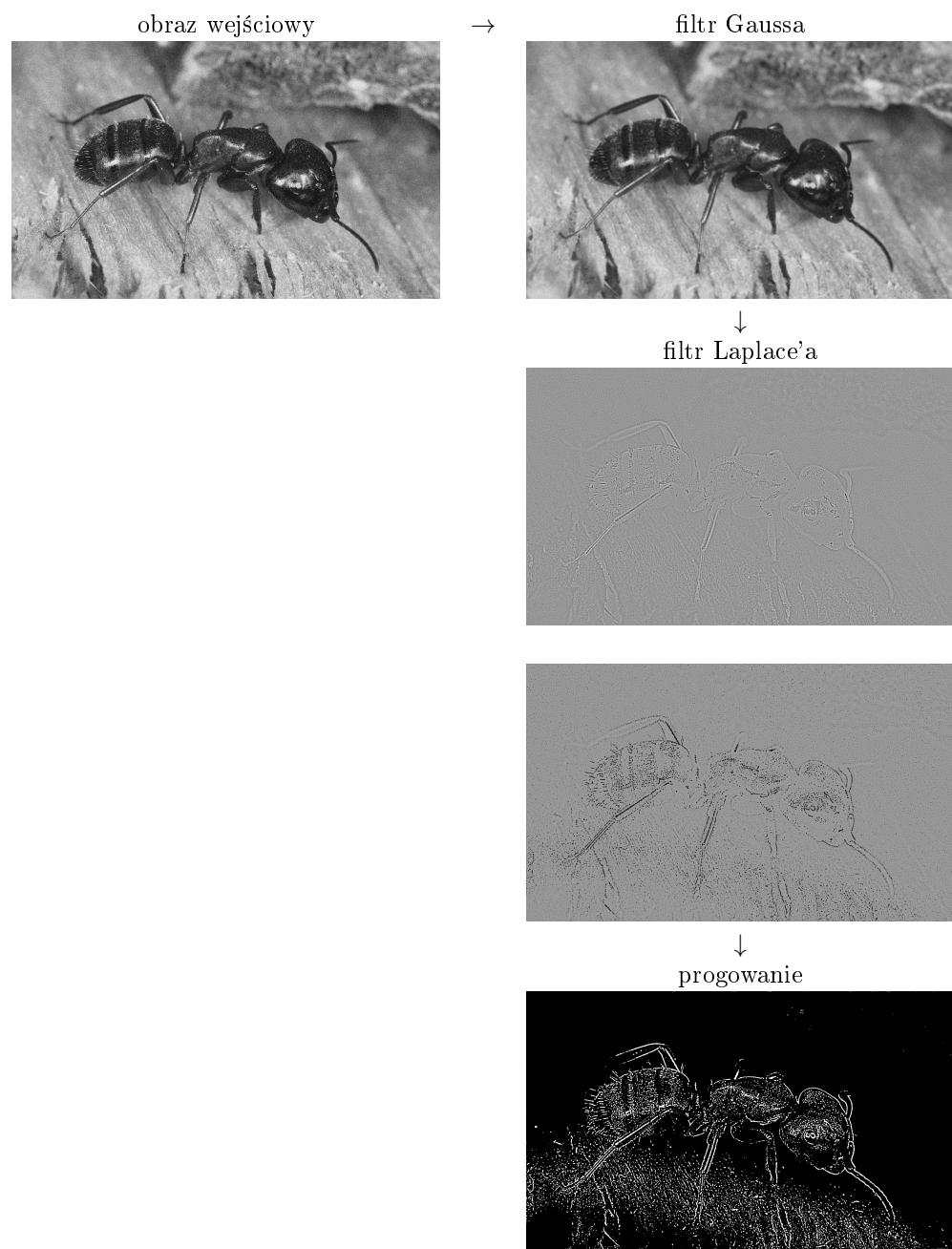
13.2 Poprawa konturu

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

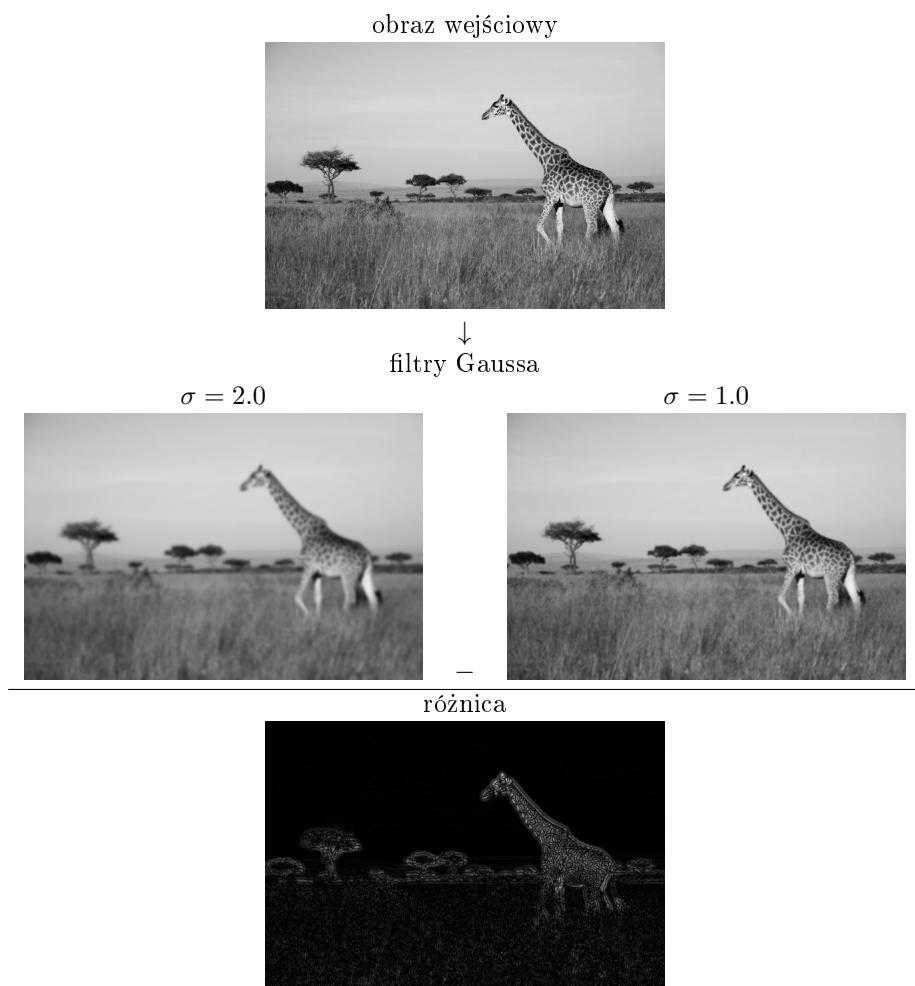
Gradient

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.

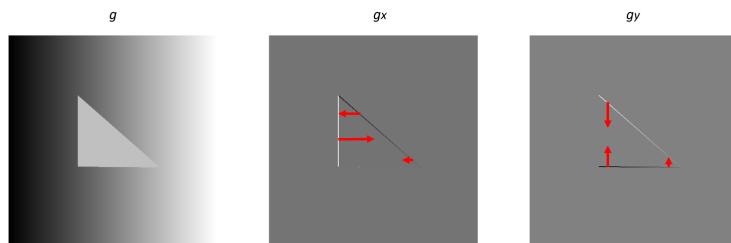
Wartość gradientu G oblicza się jako:

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

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

Kierunek gradientu R oblicza się ze wzoru:

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

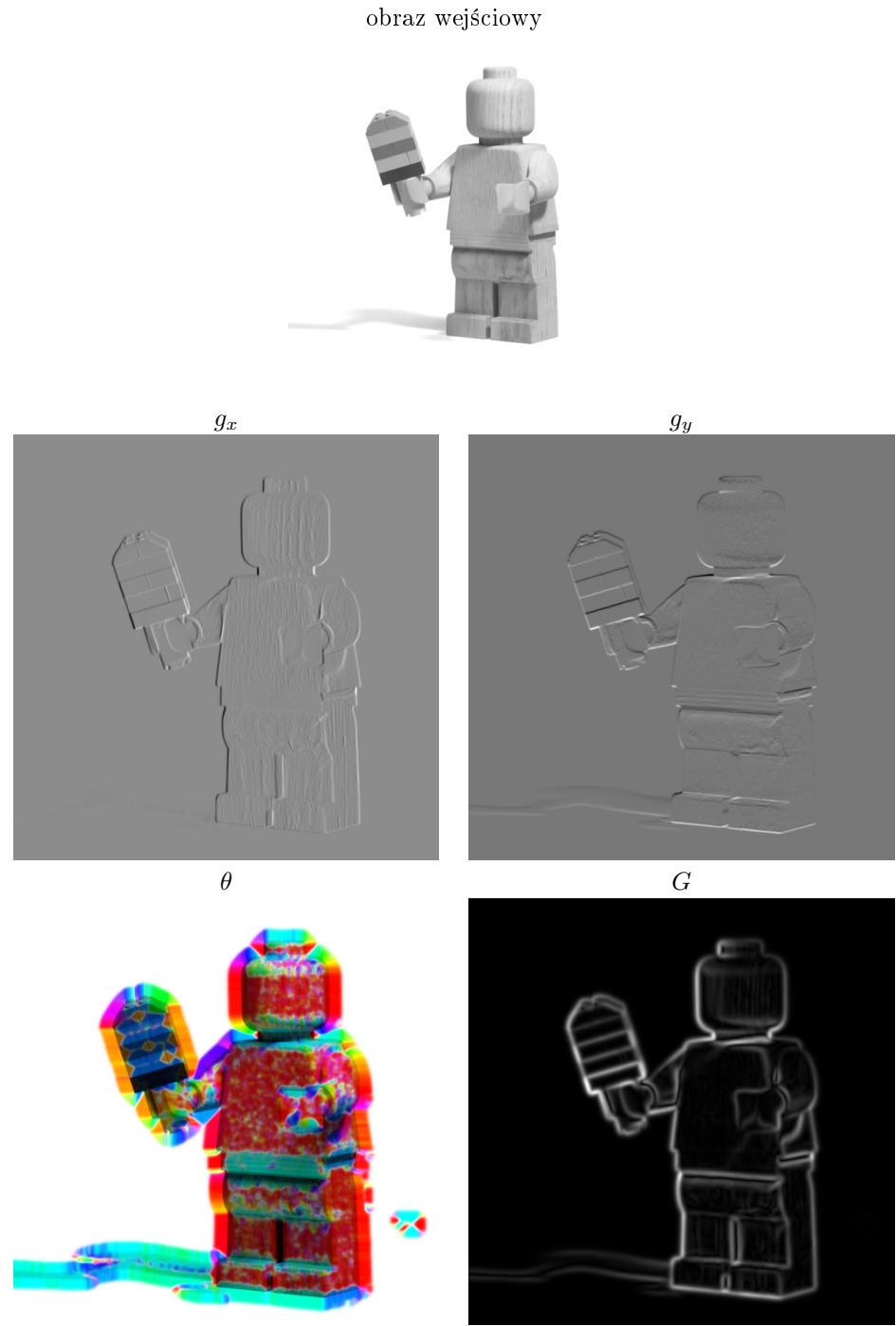


Przykład: gradient wyznaczony przez operatory Sobel'a → grafika 13.16

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

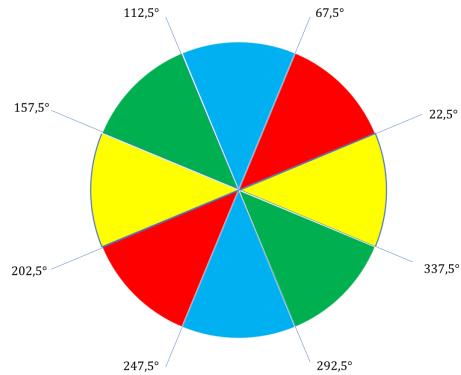
- W wielu metodach wykrywania krawędzi stosuje się stały próg, aby zdecydować, czy w tym miejscu znajduje się krawędź, czy nie. W zależności od kształtu krawędzi tworzy to większe obszary, na których może znajdować się ewentualna krawędź.
- W przypadku tłumienia wartości niemaksymalnych gradientu jego 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° (w kierunku poziomym), 45° (wzdłuż dodatniej przekątnej), 90° (w kierunku pionowym) lub 135° (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

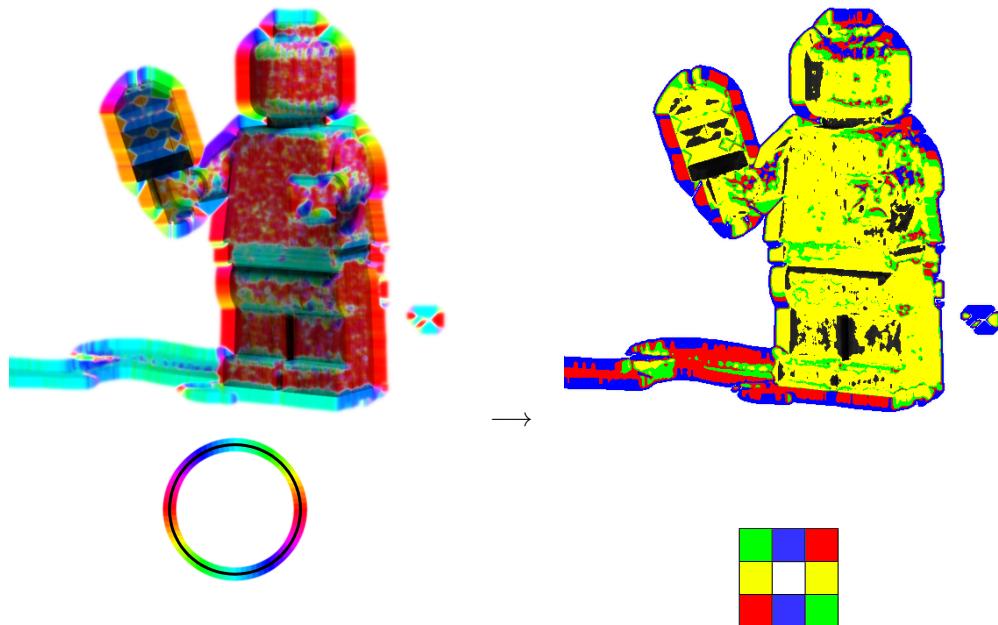


Grafika 13.16: Gradient wyznaczony przez operatory Sobel'a.

- Możliwe przedziały dla wartości kierunku gradientu θ :



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

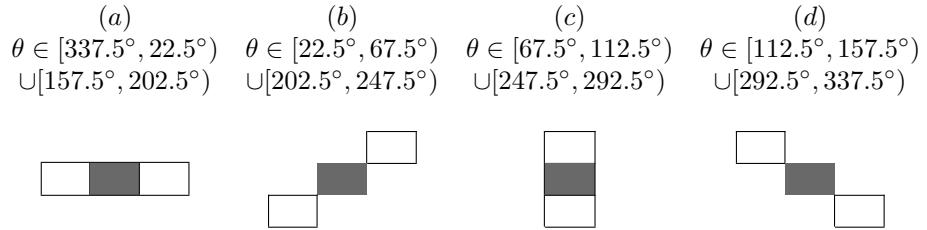


Grafika 13.17: Gradient wyznaczony przez operatory Sobel'a i przypodziałkowanie możliwych kierunków krawędzi.

Algorytm Non-Maximum Suppression

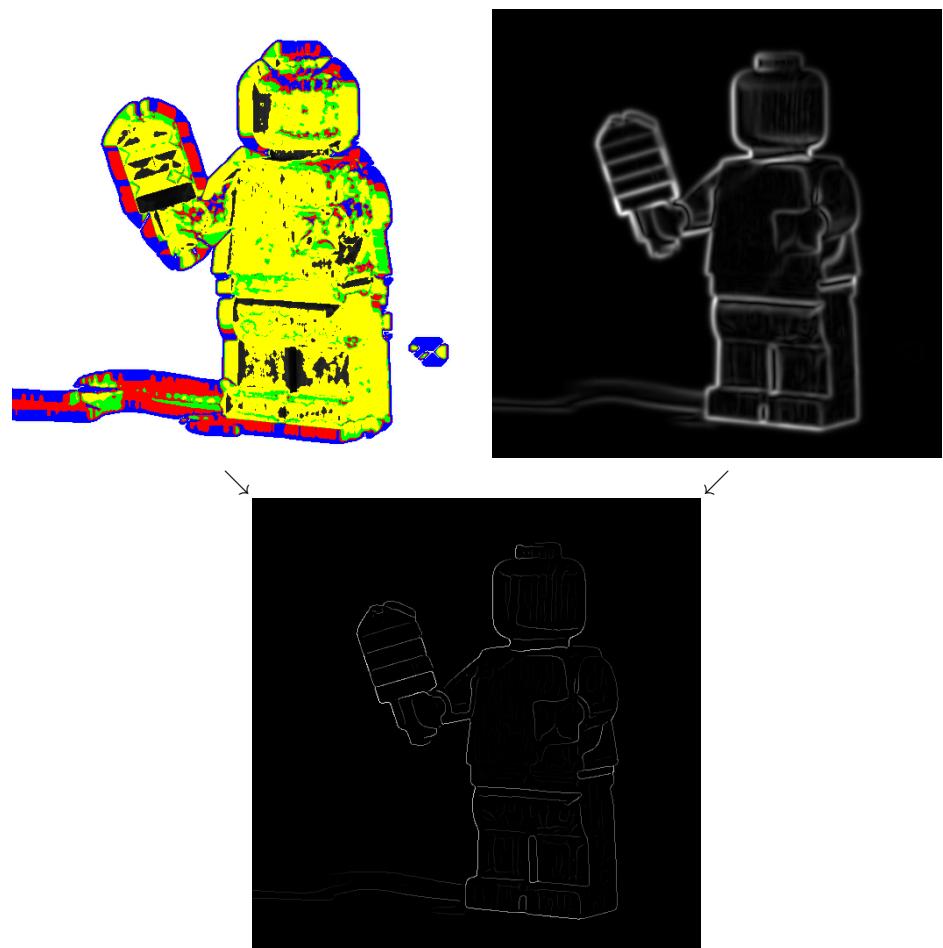
- Wybór dwóch sąsiadów (sąsiedztwo 8-spójne) aktualnego piksela, tak aby spełniony był warunek "*w poprzek do kierunku krawędzi*".

Temu celowi służą poniższe maski. Wybór maski zależy od kierunku gradientu θ bieżącego piksela:



- Jeżeli odchylenia kierunków gradientu aktualnego piksela i jego dwóch sąsiadów przekraczają daną wartość progową:
→ przejście do następnego aktualnego piksela i rozpoczęcie od pkt. 1.
- 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.
- 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).

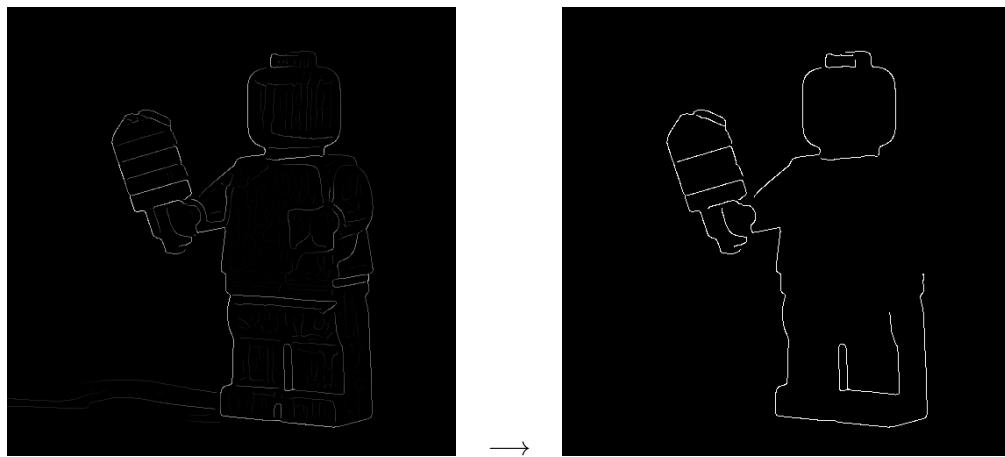
13.2.2 Progowanie z histerezą (*Hysteresis-Threshold*)

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
 - $T_1 = \frac{1}{2}T_2$

Przykład: progowanie z histerezą → grafika 13.19



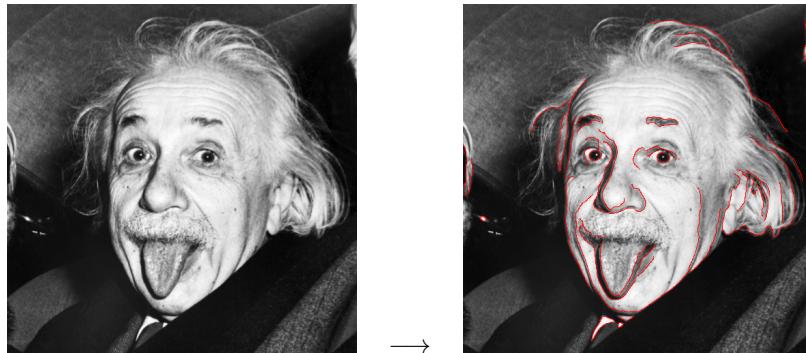
Grafika 13.19: Progowanie z histerezą

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 13.20



Grafika 13.20: 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 13.21

1. Redukcja szumu

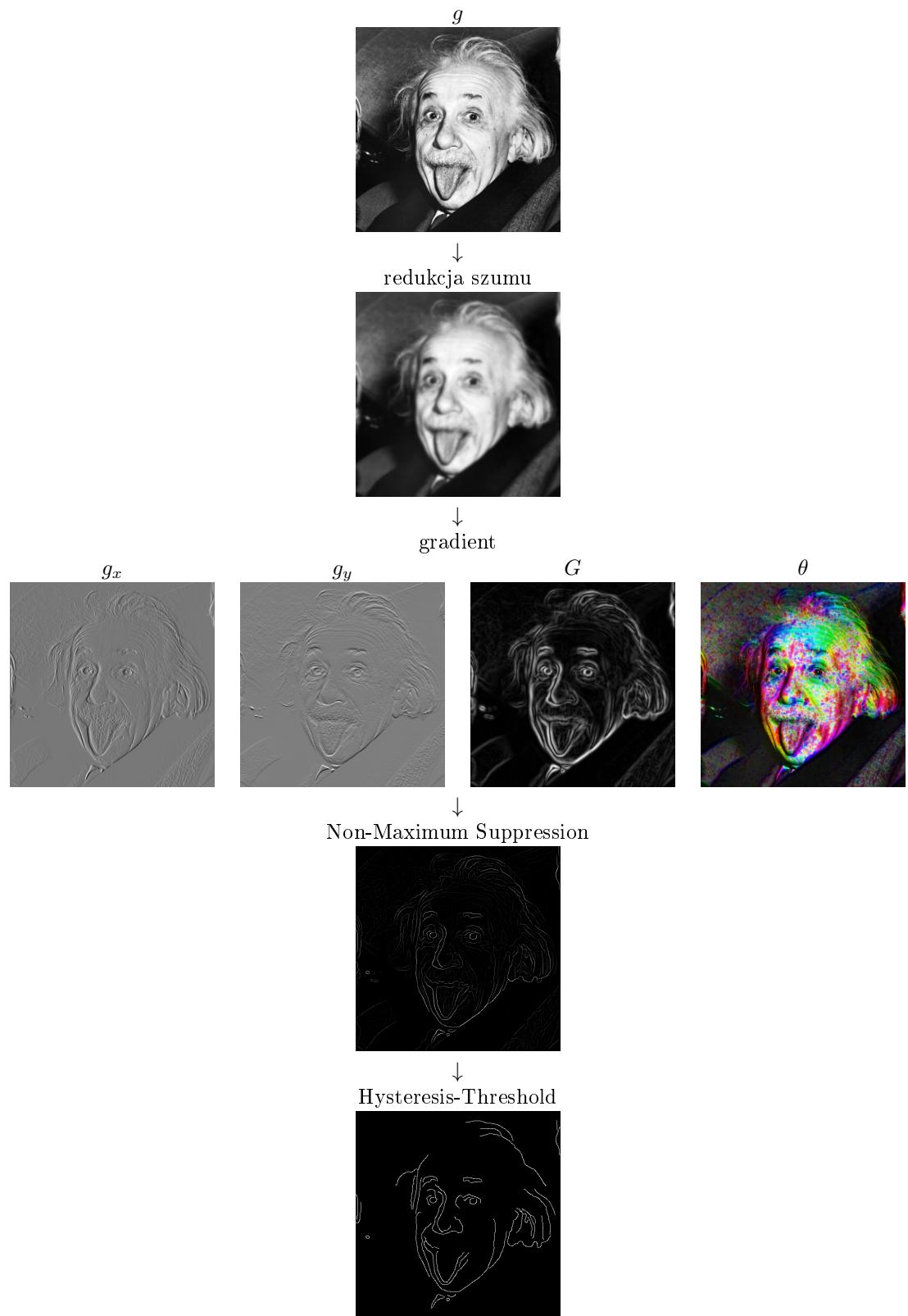
- filtr Gaussa
- filtr o współczynnikach dwumianu Newtona

2. Wyznaczenie gradientu obrazu

- operator Robertsa
- operator Prewitta
- operator Sobel'a

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

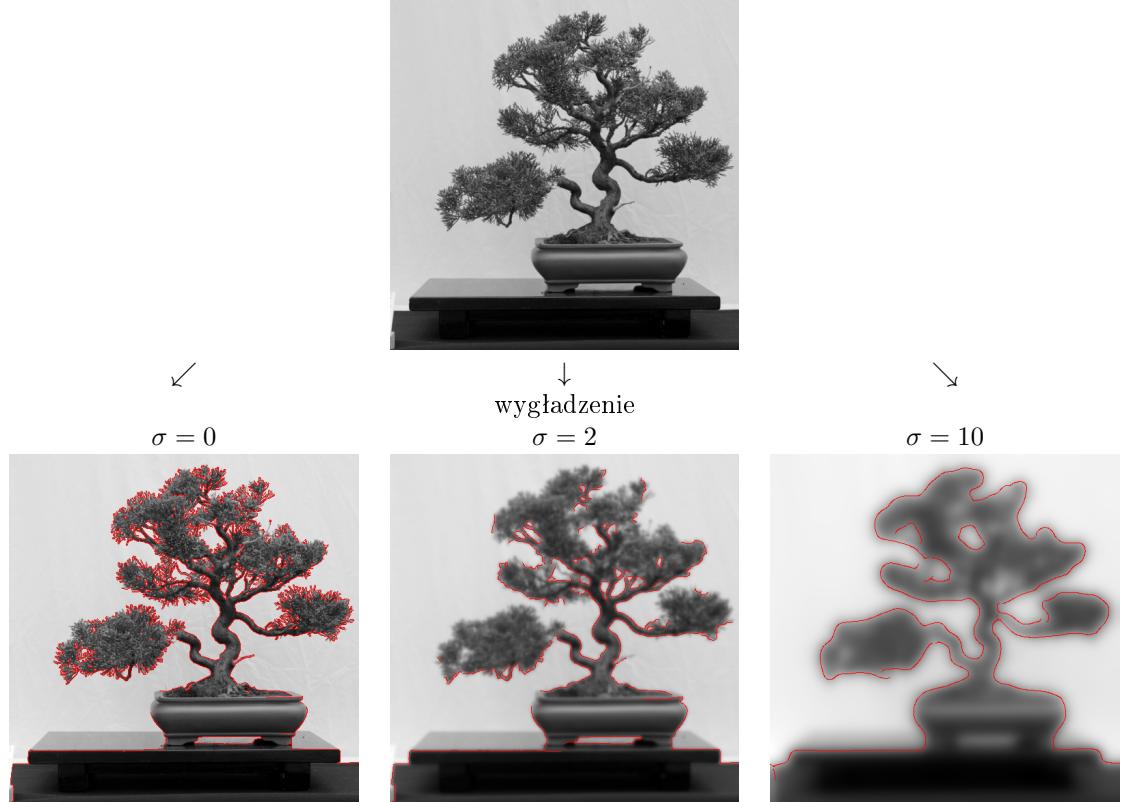
3. Usunięcie niemaksymalnych pikseli (Non-Maximum Suppression)
4. Progowanie z histerezą (Hysteresis-Threshold)



Grafika 13.21: Algorytm wykrywania krawędzi Canny'ego.

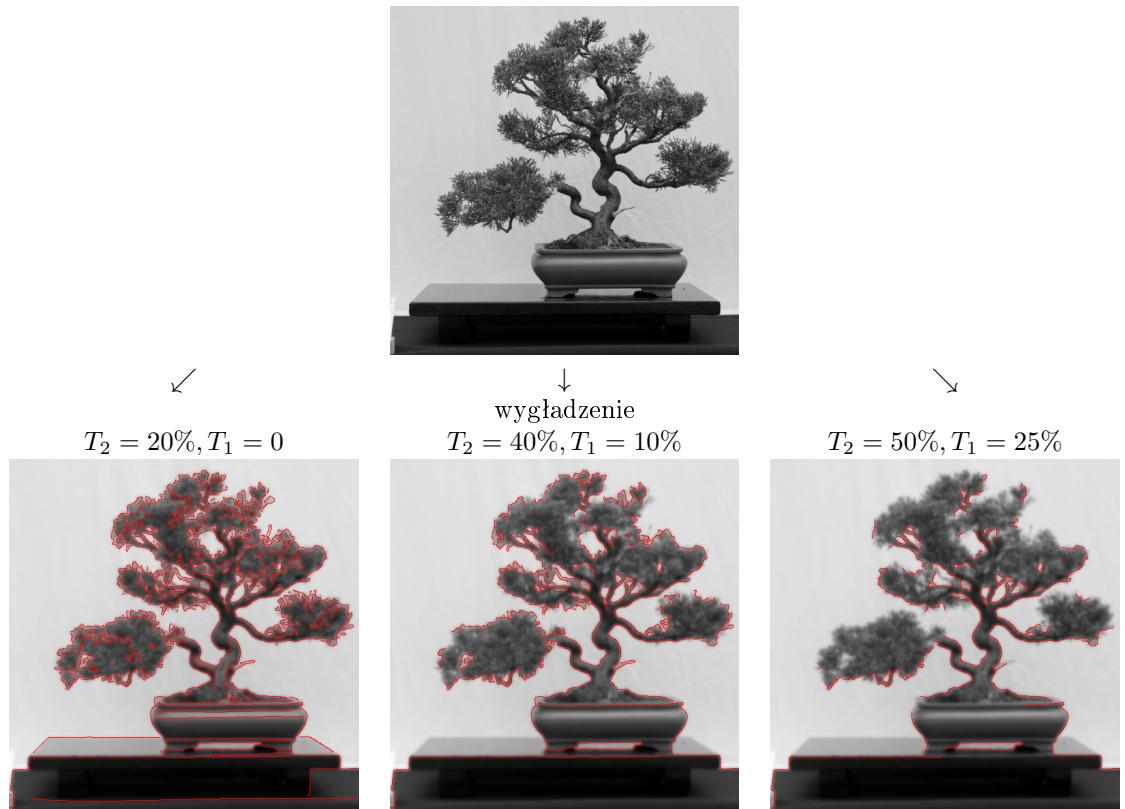
Detektor krawędzi Cannyego: wybór parametrów

- wygładzenie (redukcja szumu)
→ grafika 13.22



Grafika 13.22: Detektor krawędzi Canny'ego- wybór parametrów wygładzenia filtrem Gaussa (progowanie z histerezą $T_2 = 50\%$, $T_1 = 25\%$).

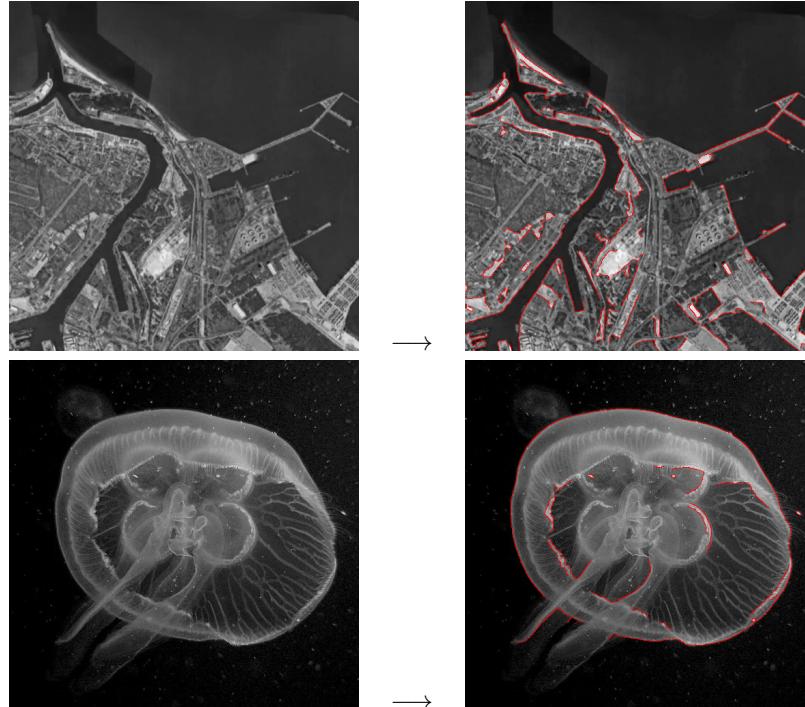
- progowanie z histerezą
→ grafika 13.23



Grafika 13.23: Detektor krawędzi Canny'ego- wybór parametrów progowania (wygładzenie filtrem Gaussa $\sigma = 2$).

Problemy klasycznych metod wykrywania krawędzi:

1. wykrywanie cienkich linii (struktur) w obrazie - "grzbiety i doliny" (dwie lokalnie równoległe krawędzie) —> grafika 13.24



Grafika 13.24: Detektor krawędzi Canny'ego- problemy.

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.

Mozliwe operatory punktowe wyznaczające odległości kolorów to

- norma L_1 (suma wartości),
- norma L_2 (odległość euklidesowa),
- norma L_∞ (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 13.25

Przykład 2

→ grafika 13.26

Przykład 3

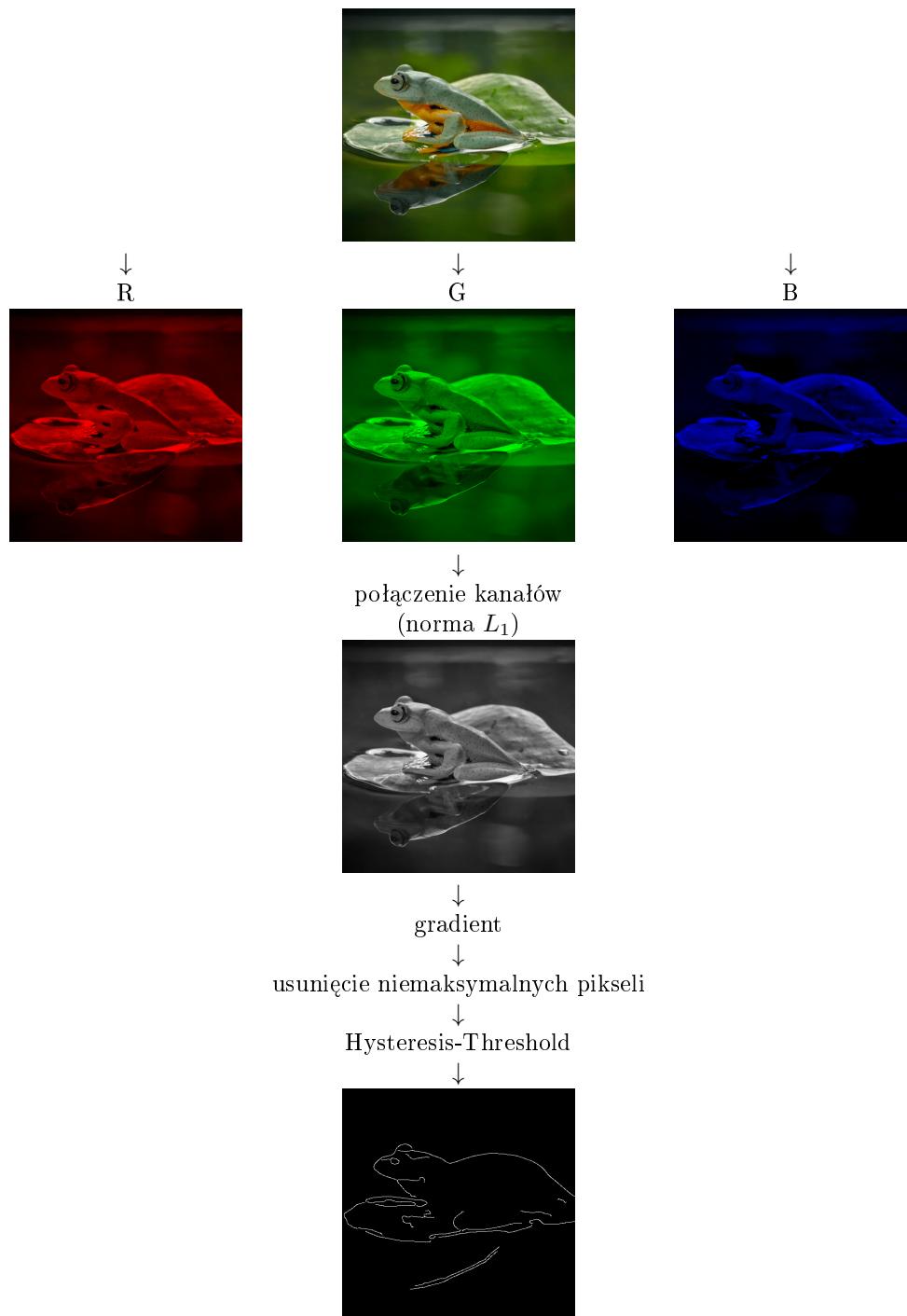
→ grafika 13.27

Zalecana w literaturze kolejność kroków algorytmu i wybór operatora:

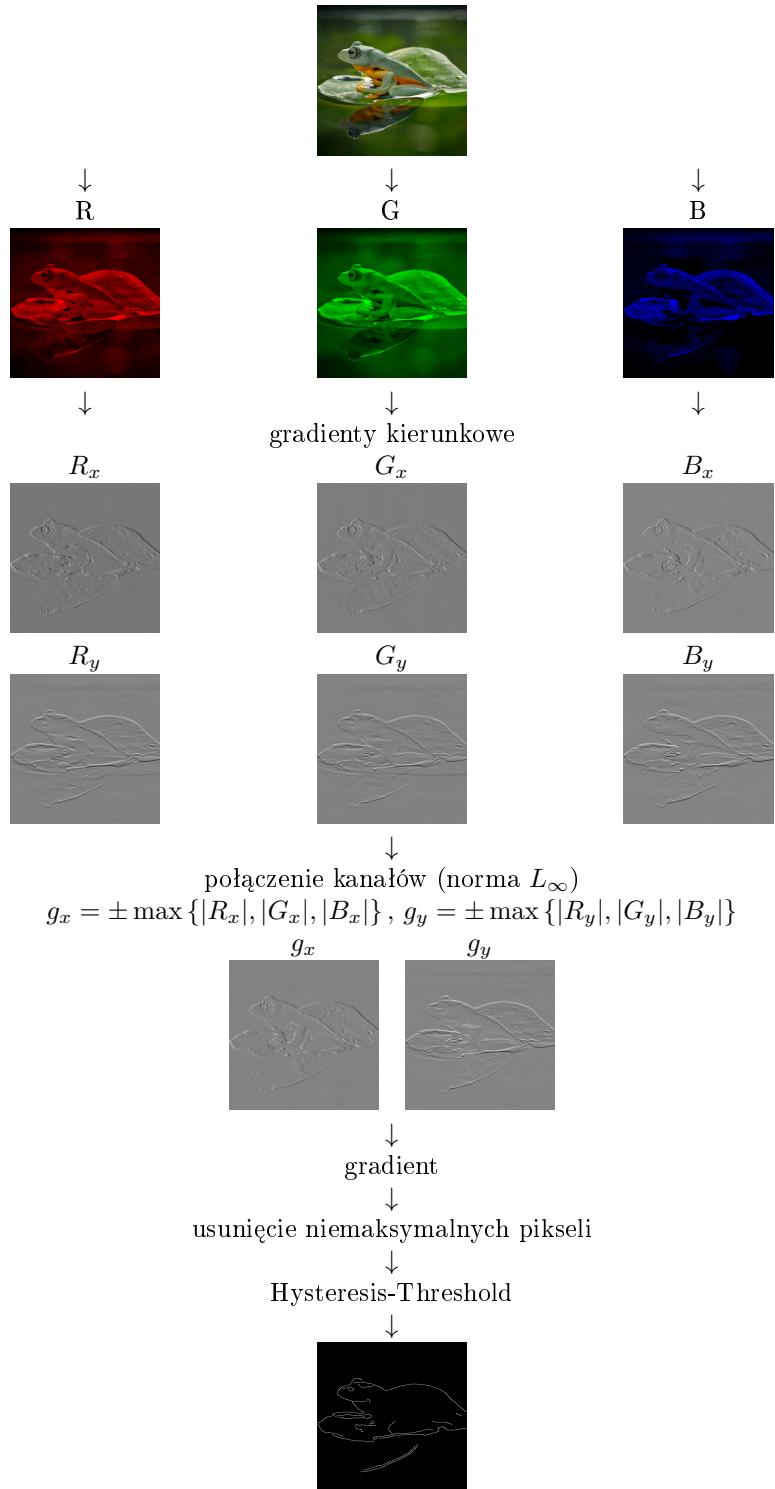
- obliczenie gradientu niezależnie dla każdego kanału,
- wybór operatora zgodnie z normą L_∞ (maksimum wartości).

Wyniki:

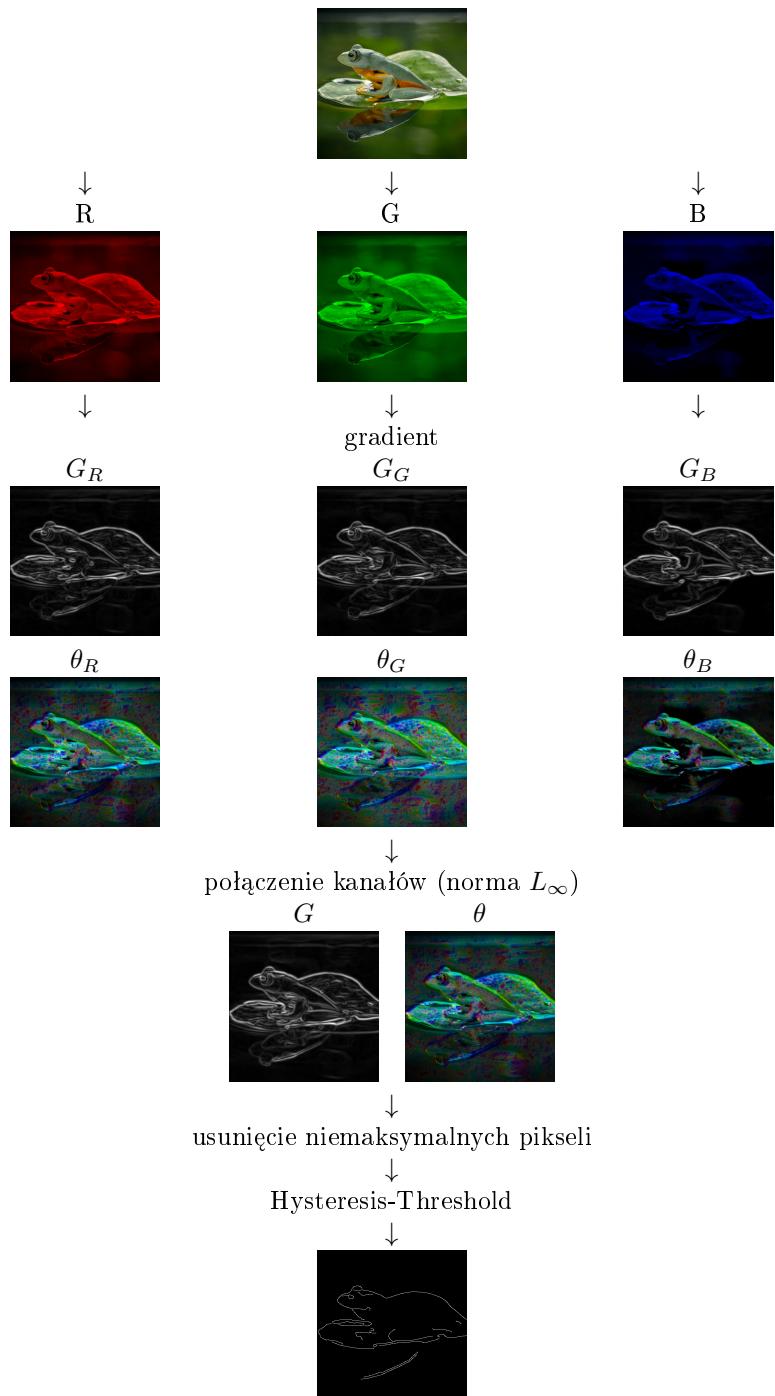
- krawędzie RGB są wyraźniejsze niż dla obrazów w odcieniach szarości,
- większość krawędzi (ponad 90%) jest identyczna.



Grafika 13.25: Algorytm wykrywania krawędzi Canny'ego dla obrazów RGB.



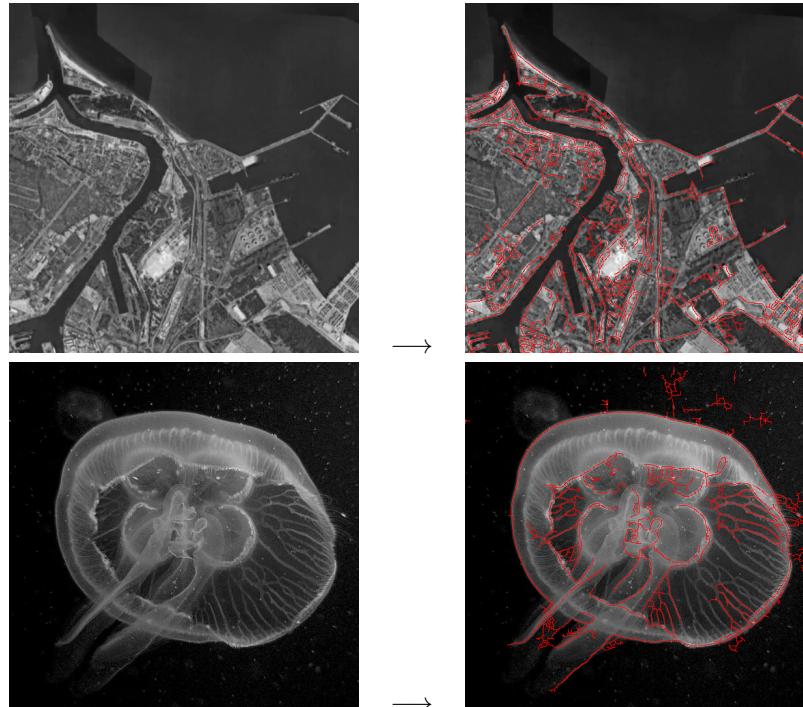
Grafika 13.26: Algorytm wykrywania krawędzi Canny'ego dla obrazów RGB.



Grafika 13.27: Algorytm wykrywania krawędzi Canny'ego dla obrazów RGB.

13.5 Detekcja grzbietów (*Ridge Detection*)

→ grafika 13.28



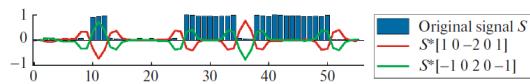
Grafika 13.28: Detektor grzbietów w obrazie

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 13.29

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 równoległy.

- 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.

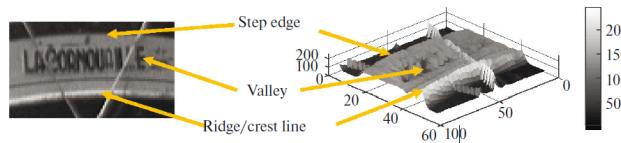


Illustration of features in images by elevation of the image intensity.

Grafika 13.29: 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

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

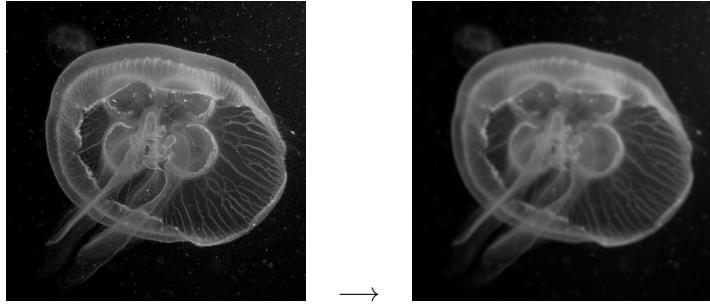
Zastosowanie: ekstrema funkcji wielu zmiennych

1. Wygładzenie obrazu

filtr Gausa → grafika 13.30

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 13.30: 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 13.31

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:

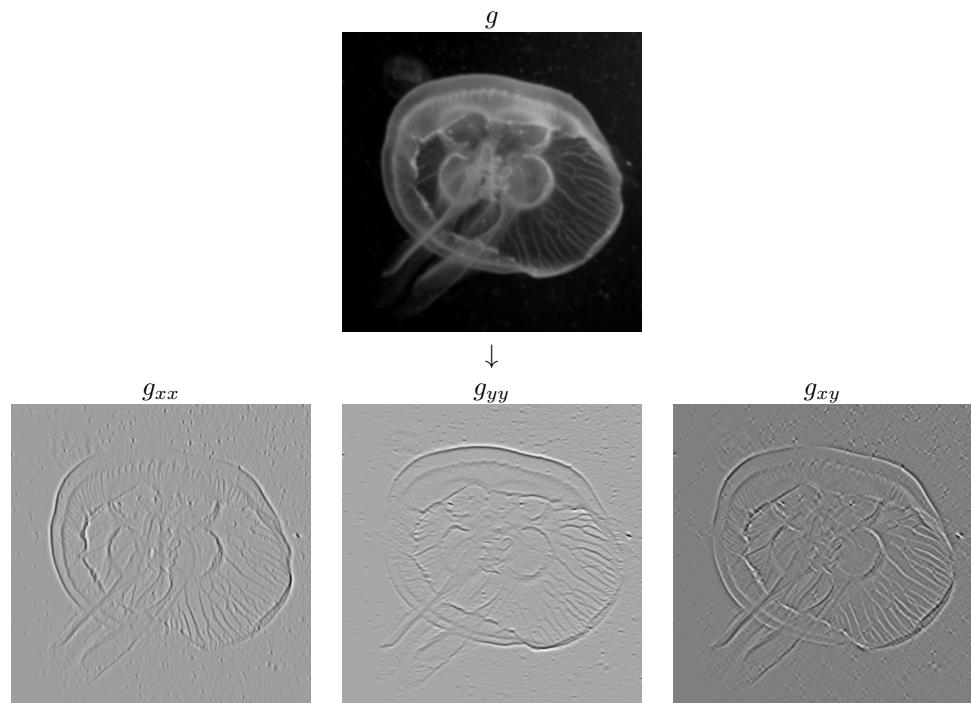
$$\begin{cases} k_1(m, n) = \frac{1}{2}(\mathcal{H}_{11} + \mathcal{H}_{22}) - \frac{1}{4}\sqrt{(\mathcal{H}_{11} + \mathcal{H}_{22})^2 + 4\mathcal{H}_{12}^2} \\ k_2(m, n) = \frac{1}{2}(\mathcal{H}_{11} + \mathcal{H}_{22}) + \frac{1}{4}\sqrt{(\mathcal{H}_{11} + \mathcal{H}_{22})^2 + 4\mathcal{H}_{12}^2}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

→ grafika 13.32

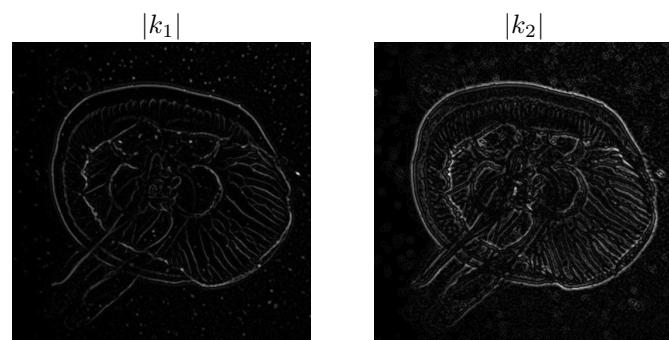
4. Wyznaczenie krzywizny obrazu

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)



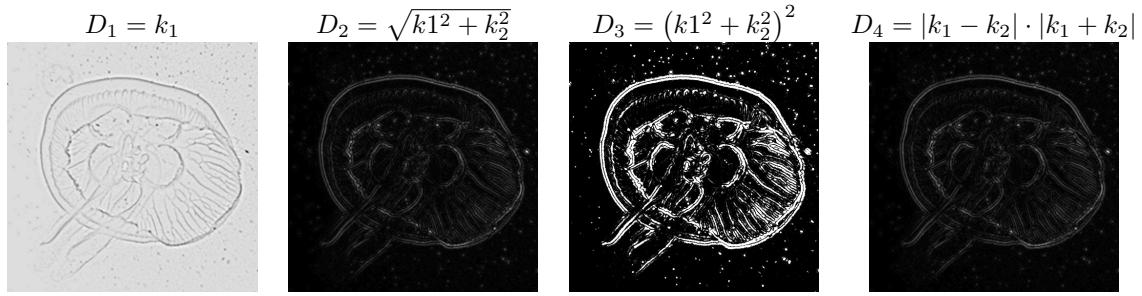
Grafika 13.31: Drugie pochodne kierunkowe wygładzonego obrazu



Grafika 13.32: 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 13.33



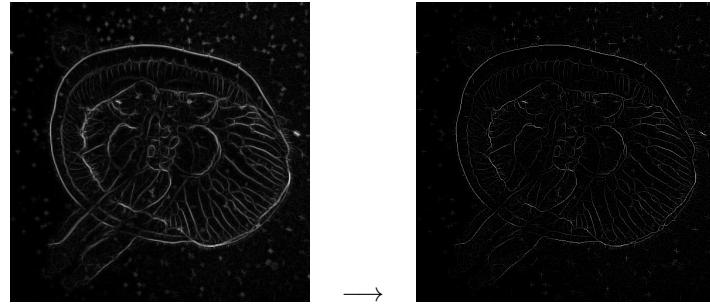
Grafika 13.33: Wyznaczenie krzywizny obrazu z wartością własnych k_1 i k_2 macierzy Hessego.

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):

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

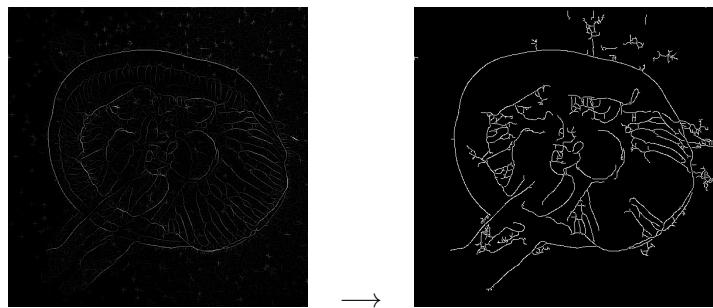
→ grafika 13.34



Grafika 13.34: Non-Maximum Suppression (usunięcie niemaksymalnych pikseli)

6. Hysteresis threshold (progowanie histerezy)

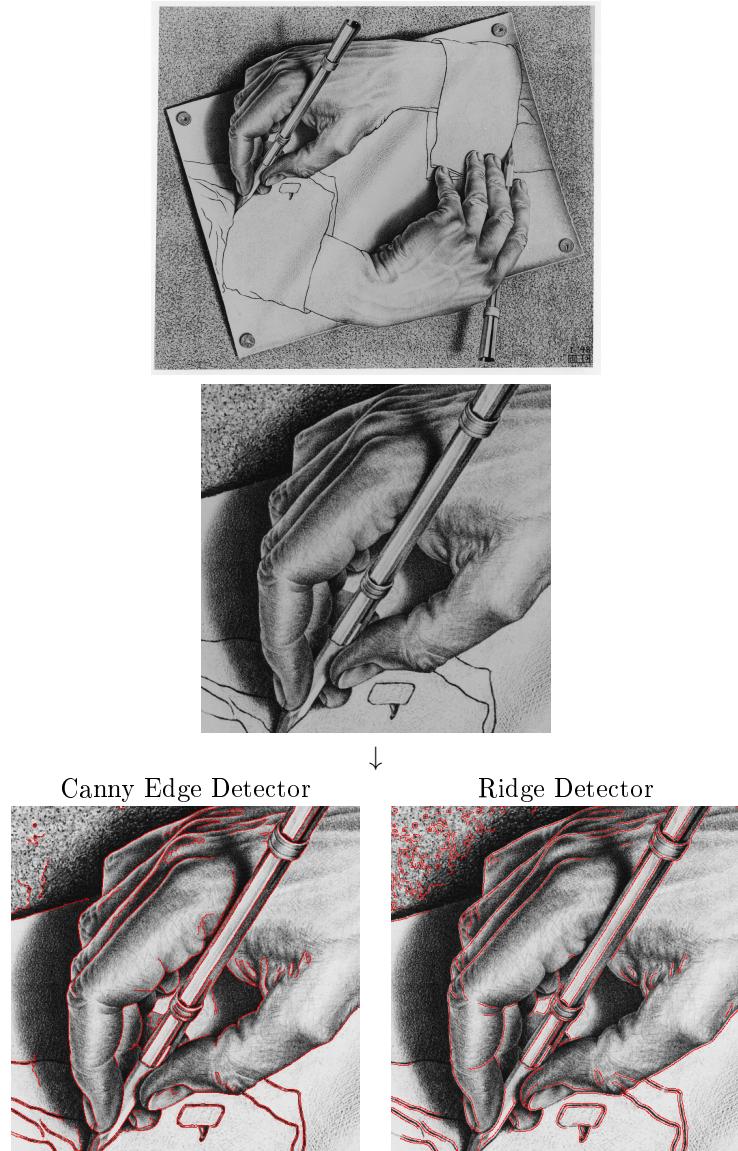
→ grafika 13.35



Grafika 13.35: Hysteresis threshold (progowanie histerezy)

Porównanie Canny Edge Detector - Ridge Detector

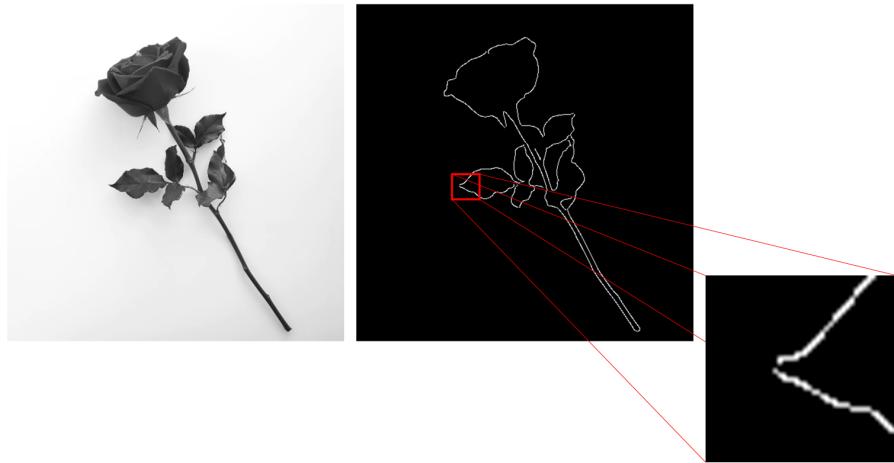
→ grafika 13.36



Grafika 13.36: Wykrywanie krawędzi i grzbietów w obrazie. Obraz wejściowy: M.C. Escher, Drawing Hands. ©2017 The M.C. Escher Company, The Netherlands.

13.6 Detekcja kątów obiektów (*Corner detection*)

→ grafika 13.37



Grafika 13.37: Detekcja krawędzi algorytmem Canny'ego - problem z lokalizacją narożnika obiektu.

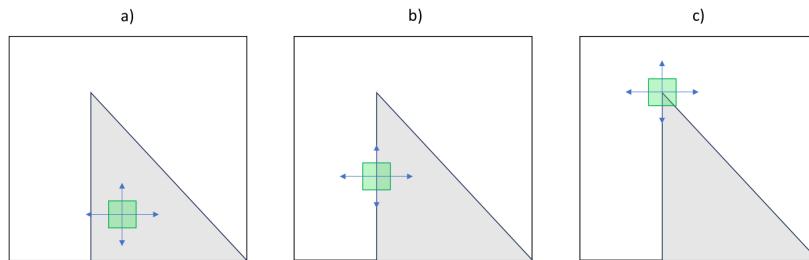
Kąt (narożnik) obiektu

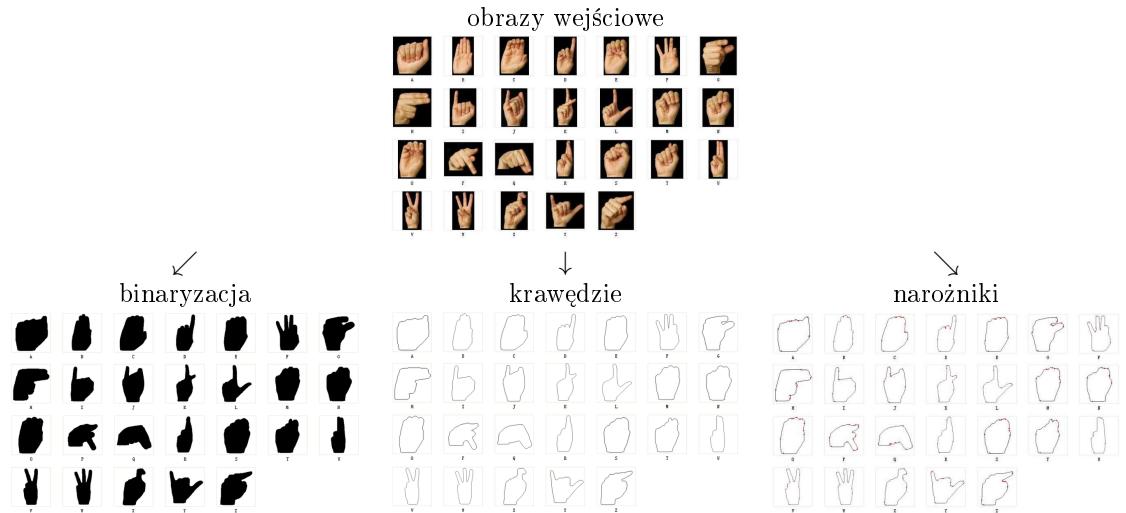
- przecięcie dwóch krawędzi
- punkt, dla którego w lokalnym sąsiedztwie występują dwa dominujące i różne kierunki krawędzi.

Przykład: Rozpoznawanie gestów w języku migowym → grafika 13.38

13.6.1 Harris Corner Detectior

Motywacja

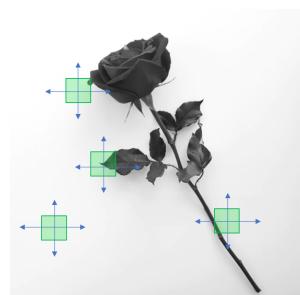




Grafika 13.38: 26 przykładowych obrazów gestów dloni dla 26 liter w amerykańskim języku migowym. Zródło: Linghua Li "Corner Detection of Hand Gesture" TELKOM-NIKA Indonesian Journal of Electrical Engineering 10(8)

- 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:

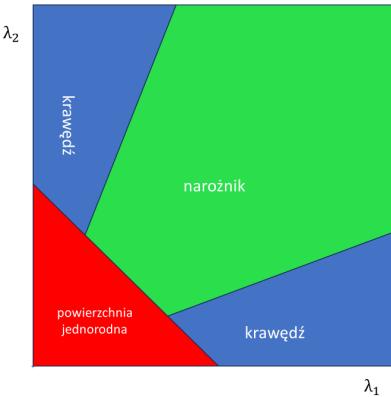
- $\text{rang}(\mathcal{M}) = 2$ punkt narożnikowy
- $\text{rang}(\mathcal{M}) = 1$ (prosta) krawędź
- $\text{rang}(\mathcal{M}) = 0$ jednorodna, niestrukturyzowana powierzchnia

- Wartości własne λ_1, λ_2 macierzy \mathcal{M} opisują struktury sąsiedztwa: λ_1, λ_2 są proporcjonalne do zmian wartości szarości w obrazie wzduż 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 λ_1, λ_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 λ_1, λ_2 macierzy \mathcal{M} , stosuje się następujące zależności dla wyznacznika (\det) i śladu macierzy (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

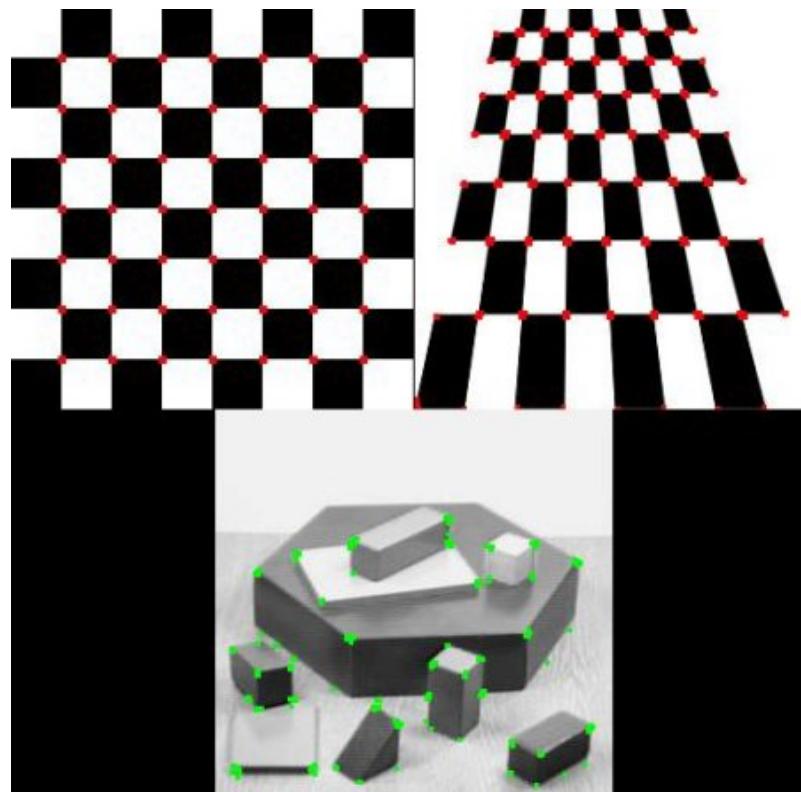
1. obliczenie pochodnych kierunkowych g_x i g_y obrazu
 \rightarrow filtry gradientowe, operator Sobel'a,...
2. obliczenie iloczynów pochodnych dla każdego piksela

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

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 13.39



Grafika 13.39: Detekcja narożników algorytmem Harris'a. Źródło: Open Source Computer Vision, Harris Corner Detection