

Marek SAMKO

Uniwersytet Warmiński – Mazurski, Wydział Matematyki i Informatyki

DETEKCJA KRAWĘDZI W CYFROWYCH OBRAZACH MONOCHROMATYCZNYCH – PRZEGLĄD ROZWIĄZAŃ

Streszczenie. W artykule przedstawiony jest przegląd metod detekcji krawędzi w obrazach monochromatycznych. Zawiera opis detektorów gradientowych, detektorów drugiej pochodnej oraz algorytmu Canny’ego.

Słowa kluczowe: detekcja krawędzi, obrazy krawędziowe, detektory gradientowe, detektory drugiej pochodnej

EDGE DETECTION IN MONOCHROMATIC DIGITAL IMAGE – SURVEY

Summary. In this article methods of edge detection in monochromatic digital images are disclosed. It contains description of gradient detectors, second derivative detectors and Canny’s algorithm.

Keywords: edge detection, edge images, gradient detectors, second derivative detectors

1. Wstęp

Detekcja krawędzi jest jednym z podstawowych elementów przetwarzania obrazów i jest wstępnym krokiem w wielu operacjach z tej dziedziny. Jest ona ważnym etapem wykonywanym w wielu aplikacjach (takich jak systemy segmentacji obrazu, rozpoznawania czy analizy scen) ze względu na to, że krawędzie niosą bardzo istotne informacje o obiektach znajdujących się na obrazie. Badania ludzkiego sposobu postrzegania obrazów wykazały, że opiera się on na wcześniejszym wyodrębnieniu krawędzi obiektów, a następnie na rozpoznawaniu i identyfikacji tych obiektów na podstawie zaobserwowanych krawędzi. Dlatego też proces

wykrywania krawędzi znalazł zastosowanie w komputerowych systemach analizy i rozpoznawania obrazów (*Computer Visual System*). Ponadto, proces detekcji krawędzi służy uproszczeniu analizy obrazu przez redukcję wielkości danych, jakie będą przetwarzane, jednocześnie zachowując użyteczną informację o strukturze granic między obiektami. Detekcja krawędzi ma szerokie zastosowanie m.in. w medycynie, przy analizie różnego rodzaju zdjęć, tomografii komputerowej, w systemach zabezpieczeń (np. rozpoznawanie odcisków palców czy siatkówki oka), w analizie zdjęć lotniczych i satelitarnych oraz we wszelkich komputerowych systemach wizyjnych (np. rozpoznanie obiektów znajdujących się w otoczeniu robota).

W artykule przedstawione są główne etapy detekcji krawędzi. Omówione są ich własności i warunki, jakie operacje te muszą spełniać. Dalej przedstawione są dwie podstawowe grupy metod detekcji krawędzi: metody gradientowe i metody przejść przez zero (oparte na drugiej pochodnej funkcji obrazu). Metody gradientowe wyznaczają pierwszą pochodną funkcji obrazu. Algorytmy te są algorytmami kierunkowymi – umożliwiają one wykrycie krawędzi przebiegających w pewnym wybranym kierunku. W celu otrzymania wszystkich krawędzi konieczne jest zastosowanie algorytmu w dwóch prostopadłych kierunkach. W przypadku algorytmów przejść przez zero, które są algorytmami bezkierunkowymi, wynikiem jest zbiór krawędzi niezależny od kierunku ich przebiegu. Przedstawione przykłady działania operatorów będą dotyczyły obrazka przedstawionego na rys. 1.



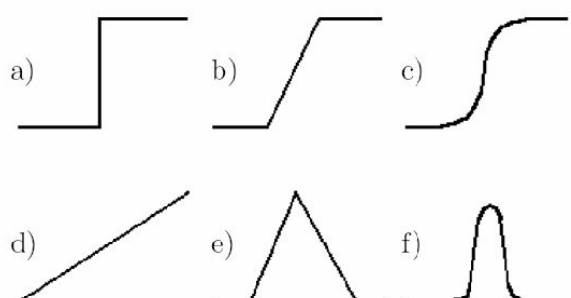
Rys. 1. Obraz źródłowy
Fig. 1. Source image

2. Krawędzie w obrazach: ich detekcja i typy

Detekcja krawędzi jest procesem wydobywania z obrazu pewnych jego cech charakterystycznych – nagłych zmian funkcji jasności. Zmiany te powstają na granicach różnych obiektów lub ich fragmentów uwidocznionych na obrazie, a wynikają z: różnego oświetlenia poszczególnych fragmentów sceny, różnego współczynnika odbijania/pochłaniania/przepuszczenia powierzchni czy tekstury obiektów. Najczęściej spotykanymi rodzajami zmian funkcji obrazu, odpowiadających krawędziom, są: krawędzie uskokowe (skokowe), linie oraz różnego rodzaju przecięcia powstające przez zetknięcie się przynajmniej dwóch krawędzi.

Celem detekcji jest wykrycie zmian funkcji obrazu i ich zidentyfikowanie. W tym celu wyznaczane są pochodne funkcji obrazu. Z powodu dużej wrażliwości wyznaczonych pochodnych na zaszumienie powstające na etapie pozyskiwania obrazu, jego kwantyzacji i dyskretyzacji stosowane jest wygładzanie obrazu. Operacja ta skutkuje jednak utratą części informacji o obiektach uwidocznionych na obrazie i ich lokalizacji.

Dla obrazów monochromatycznych krawędź definiuje się zazwyczaj jako linię (w najprostszym przypadku prostą) oddzielającą obszary o różnej wartości funkcji intensywności obrazu [1]. Krawędzie pojawiają się więc na granicy obszarów o różnych wartościach jasności (różnych poziomach szarości). Krawędzie znajdujące się na obrazie to nie tylko kontury obiektów oraz granice między różnymi obiektami, ale także różnice w jasności pikseli spowodowane odbiciami światła, cieniami oraz zastosowaniem różnych tekstur. Można wyróżnić kilka rodzajów krawędzi ze względu na kształt przejścia funkcji obrazu z jednego obszaru do drugiego. Kilka najważniejszych przedstawia rys. 2.



Rys. 2. Profile poprzeczne krawędzi: a) skokowy nagły, b) skokowy narastający, c) skokowy wygładzony, d) liniowy, e) liniowy typu „dach”, f) liniowy typu „przeszkoda”

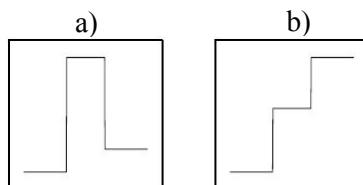
Fig. 2. Edges cross – sections: a) sharp step, b) sloped step, c) smooth step, d) linear, e) linear “roof”, f) linear “barrier”

Najczęściej spotykanymi rodzajami krawędzi są uskoki, linie i węzły.

2.1. Detekcja krawędzi

Krawędzie uskokowe, zwane też skokowymi, są najczęściej spotykanymi rodzajami krawędzi i stanowią model teoretyczny opisu detektorów. Krawędzie te pojawiają się pomiędzy dwoma fragmentami obrazu mającymi mniej więcej stałą, ale różniącą się między sobą poziom jasności. Krawędzie uskokowe umiejscawiane są w dodatnim maksimum lub ujemnym minimum pierwszej pochodnej lub w miejscu przejścia przez zero drugiej pochodnej funkcji obrazu. W przestrzeni dwuwymiarowej, jaką jest płaszczyzna obrazu, pierwsza pochodna funkcji jasności zdefiniowana jest przez operator gradientowy. Natomiast druga pochodna aproksymowana jest przez laplasjan lub operator drugiej pochodnej w kierunku gradientu. Idealna krawędź uskokowa jest rzadko spotykana, częściej uskoki pojawiają się w grupach

(np. podwójne krawędzie uskokowe). Najczęściej spotykane są dwa rodzaje takich krawędzi: „impuls” i „schody” (rys. 3).



Rys. 3. Rodzaje podwójnych krawędzi uskokowych: a) „impuls”, b) „schody”
Fig. 3. Kinds of twin step edges: a) “impulse”, b) “stairs”

Prostym modelem matematycznym krawędzi uskokowej (nagle przejście z obszaru o jednym poziomie jasności do obszaru o innym poziomie, bez poziomów pośrednich) jest postać skoku jednostkowego w zerze:

$$u(t) = \begin{cases} 1, & \text{gdy } t > 0 \\ \frac{1}{2}, & \text{gdy } t = 0 \\ 0, & \text{gdy } t < 0 \end{cases} \quad (1)$$

jest więc całką jednowymiarowej delty Diraca:

$$u(t) = \int_{-\infty}^t \delta(\tau) d\tau$$

Można przyjąć, że krawędź leży na prostej o równaniu: $y = ax + b$, nachylonej do osi x pod kątem α . Podstawiając do wzoru prostej: $a = \operatorname{tg} \alpha$; $\rho = b \cos \alpha$, otrzymujemy postać kanoniczną:

$$x \sin \alpha - y \cos \alpha + \rho = 0 \quad (2)$$

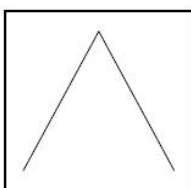
Jeżeli krawędź leżąca na prostej (2) jest krawędzią uskokową opisaną wzorem (1), wartość funkcji obrazu w punkcie (x, y) można opisać wzorem:

$$f(x, y) = z_1 + (z_2 - z_1)u(x \sin \alpha - y \cos \alpha + \rho) \quad (3)$$

gdzie: z_1 i z_2 – jasności obszarów, które rozdziela badana krawędź.

2.2. Detekcja linii

Linie powstają w wyniku różnego oświetlenia obiektów znajdujących się blisko siebie lub przy obiektach znajdujących się na tle innych. Profil linii przedstawia rys. 4.



Rys. 4. Profil krawędzi liniowej
Fig. 4. Linear edge cross – section

Jak widać, krawędzi liniowej odpowiada ekstremum funkcji obrazu. Krawędź tego typu wyznaczana jest jako przejście przez zero pierwszej pochodnej funkcji obrazu lub lokalne maksimum laplasjanu.

2.3. Detekcja przecięć

Przecięcia krawędzi występują np. w przypadku, gdy jeden obiekt przykrywa inny. Ze względu na różne ułożenie krzyżujących się krawędzi można wyróżnić kilka typów przecięć: np. typu T, L, Y, X.

Pojęcie „krawędź” jest stosowane do wszystkich wyżej wymienionych rodzajów. Jednak, jak wspomniano wcześniej, najbardziej rozpowszechnione są krawędzie uskokowe i to dla nich powstała większość algorytmów detekcji.

3. Rodzaje detektorów i zasada ich działania

3.1. Rodzaje detektorów

Detektory krawędzi są urządzeniami realizowanymi sprzętowo lub programistycznie, które służą wykryciu i zaznaczeniu zmian funkcji obrazu odpowiadających krawędziom. Na wejściu detektory otrzymują cyfrowy obraz, a na ich wyjściu pojawia się mapa krawędzi występujących na obrazie wejściowym. Niektóre detektory tworzą mapę krawędzi, zawierając w niej informację tylko o miejscu przebiegu krawędzi, inne uwzględniają także informację o mocy krawędzi (wielkość różnicy między poziomami szarości, które krawędź rozdziela) oraz jej skali. Ze względu na użycie detektorów w różnych systemach wizyjnych można wyróżnić dwie główne grupy detektorów:

- Detektory, niedysponujące wiedzą *a priori* o scenie, która ma być podana na ich wejście (informacja taka nie jest przekazywana przez inny element systemu wizyjnego ani z zewnątrz). Takie detektory mogą być zastosowane do różnych obrazów, niezależnie od ich treści. Do działania używają jedynie informacji o lokalnym sąsiedztwie pikseli obrazu wejściowego.
- Detektory kontekstowe, które dysponują wiedzą o strukturze i treści sceny przedstawionej na obrazie wejściowym (wiedza ta przekazywana jest z zewnątrz lub przez inny element systemu wizyjnego). Zaletą takich detektorów jest lepsze działanie dla ściśle określonego rodzaju obrazów. Wadą natomiast to, że dla obrazów o innych cechach niż te, do których wykrycia detektor został przystosowany, rezultaty działania są złe.

Pierwszy rodzaj detektorów jest stosowany w ogólnych systemach wizyjnych, na których wejście podawane są różnorodne obrazy, natomiast detektory kontekstowe są używane

w systemach wyspecjalizowanych do detekcji konkretnego rodzaju obrazów i operowania na podobnych scenach (np. systemy rozpoznające odciski palców, siatkówkę oka itp.).

Działanie detektorów krawędzi jest realizowane w trzech etapach: różniczkowania obrazu, wygładzania i etykietowania krawędzi. Różniczkowanie wykonywane jest w celu wykrycia w obrazie nagłych zmian jasności pikseli, co odpowiada występowaniu krawędzi. Wygładzanie służy redukcji zaszumienia i regularyzacji obrazu zróżniczkowanego. Etykietowanie polega na tłumieniu fałszywych krawędzi oraz zwiększeniu stosunku sygnału do szumu wyznaczonych krawędzi (czyli ich wzmocnieniu). Etykietowanie jest ostatnią operacją przeprowadzaną przez detektor, natomiast dwie pierwsze mogą być wykonywane w różnej kolejności, w zależności od cech detektora. Różniczkowanie i wygładzanie obrazu jest często realizowane jako filtrowanie obrazu różniczką filtra wygładzającego.

3.2. Wygładzanie obrazu

Wygładzanie jest redukcją zaszumienia, wadą zaś utrata pewnej części informacji niesionej przez obraz. Współzależność ta powoduje, że jednym z zadań prowadzących do zdefiniowania optymalnego detektora jest znalezienie kompromisu między zmniejszeniem zaszumienia a utratą informacji o krawędziach. Znalezieniu optymalnego kompromisu między tymi dwoma cechami wygładzania służy regularyzacja. Regularyzacja różniczkowania kontrolowana jest przez parametr regularyzacji. Parametr ten określa kompromis pomiędzy redukcją zaszumienia a zachowaniem struktury obrazu i jest nazywany skalą. Regularyzacja różniczkowania może być zapewniona przez zastosowanie filtrów regularyzacyjnych, takich jak np. funkcje Greena i Gaussa. Odpowiedzi impulsowe tych funkcji w jednym wymiarze dane są wzorami:

$$r(x) = \frac{\mu}{2\sqrt{2}} e^{-\mu|x|} \left(\cos \frac{\mu|x|}{\sqrt{2}} + \sin \frac{\mu|x|}{\sqrt{2}} \right) \quad (4)$$

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (5)$$

Pomimo że powyższe filtry zapewniają kompromis pomiędzy redukcją szumu a zachowaniem struktury obrazu, nadal problemem jest wybór odpowiedniego parametru regularyzacyjnego (skali – μ w funkcji Greena i σ w funkcji Gaussa).

Atrybutami filtrów wygładzających, które wpływają na działanie detektorów krawędzi, są: liniowość filtrów i niezmienność ze względu na rotację. Filtry nieliniowe wykazują się lepszymi własnościami redukcji różnych rodzajów szumu (np. szum impulsowy) przy jednoczesnym zachowaniu informacji o obrazie niż filtry liniowe. Mimo to w detekcji krawędzi częstsze jest użycie filtrów liniowych. Niezmienność ze względu na rotację zapewnia, że

efekt wygładzania będzie taki sam, niezależnie od kierunku przebiegu krawędzi. Warunkiem, który funkcja musi spełnić, by była niezmienna ze względu na rotację (można też powiedzieć, że funkcja jest radialnie symetryczna), jest aby jej postać biegunowa była zależna tylko od odległości biegunowej $\rho = \sqrt{x^2 + y^2}$, a niezależna od kierunku $\text{tg } \phi = y/x$. Formalnie funkcja jest niezmienna ze względu na rotację, gdy dla dowolnego $\phi: f(x, y) = f(X, Y)$, gdzie $X = x \cos \phi + y \sin \phi$ i $Y = -x \sin \phi + y \cos \phi$. Operator Gaussa jest radialnie symetryczny, podczas gdy operator Greena nie jest [9].

3.3. Różniczkowanie obrazu

Różniczkowanie to wyznaczenie pochodnych funkcji obrazu. Operator różniczkowania jest charakteryzowany przez rząd, niezmiennosc ze względu na rotację oraz przez liniowość. Rząd operatora jest określany przez rząd pochodnych cząstkowych tego operatora. Operator $O_{x,y}$ jest symetryczny radialnie, jeśli $O_{x,y} = R O_{x,y}$, gdzie R – macierz obrotu. $O_{x,y}$ jest liniowy, jeśli dla każdego dodatniego α i β i dowolnej pary funkcji $f(x,y)$ i $g(x,y)$ zachodzi równość: $O_{x,y}(\alpha f(x,y) + \beta g(x,y)) = \alpha O_{x,y}(f(x,y)) + \beta O_{x,y}(g(x,y))$. Kolejność, w jakiej mają być wykonywane wygładzanie oraz różniczkowanie obrazu, zależy od liniowości operatora różniczkowania. Operator liniowy jest operatorem łącznym i rozdzielnym względem spłotu, w związku z tym dla dwóch różniczkowalnych funkcji $I(x,y)$ i $f(x,y)$ zachodzi zależność:

$$O_{x,y}(I(x,y) * f(x,y)) = O_{x,y}(I(x,y)) * f(x,y) = I(x,y) * O_{x,y}(f(x,y)) \quad (6)$$

W związku z tym kolejność wykonywania wygładzania i różniczkowania jest obojętna. Operatory nieliniowe nie są ani łączne, ani rozdzielne względem spłotu, więc ze względu na konieczność regularyzacji wygładzanie musi być przeprowadzone przed różniczkowaniem [9].

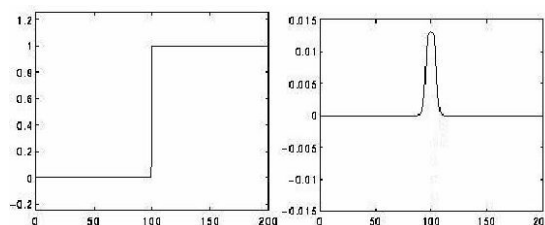
3.4. Etykietowanie krawędzi

Etykietowanie polega na lokalizowaniu krawędzi oraz zwiększaniu stosunku sygnału do szumu przez tłumienie niepoprawnie wykrytych krawędzi. Sposób lokalizowania zależy od użytego operatora różniczkowania. We wczesnych detektorach gradientowych stosowano progowanie wartości modułu gradientu. Otrzymane w ten sposób krawędzie nie miały szerokości jednego piksela. W związku z tym potrzebne było przeprowadzenie operacji szkieletyzacji. Metodę tę udoskonalono przez wprowadzenie algorytmu tłumienia punktów niemaksymalnych (*non-maximal suppression*). Działanie tego algorytmu polega na tym, że w kierunku gradientu jako piksel należący do krawędzi zaznaczany jest tylko jeden piksel o maksymalnej wartości (modułu gradientu), a nie wszystkie przekraczające podany próg.

4. Metody gradientowe

4.1. Obliczanie operatorów różniczkowych w obrazie

Jedną z grup algorytmów detekcji krawędzi są operatory gradientowe. Aby wykryć krawędź, dokonują one różniczkowania funkcji obrazu, gdyż różniczka funkcji niesie informację o natężeniu zmian funkcji różniczkowanej. W miejscu gwałtownej zmiany funkcji obrazu, a więc w miejscu znajdowania się krawędzi, różniczka funkcji będzie miała ekstremum lokalne. Dokonując progowania, możemy następnie odrzucić ekstrema o małej wartości (szum lub słabe krawędzie), a pozostawić tylko te, które odpowiadają dużym zmianom funkcji obrazu.



Rys. 5. Jednowymiarowa funkcja obrazu zawierającego krawędź skokową oraz pierwsza pochodna tej funkcji

Fig. 5. One dimensional image function for image with step edge and first derivative of this function

Gradientem funkcji obrazu nazywamy wektor:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Wyznacza się go na podstawie dwóch prostopadłych wektorów składowych w kierunku x i y :

$$\nabla_x f = \frac{\partial f}{\partial x}, \quad \nabla_y f = \frac{\partial f}{\partial y}$$

Podstawiając wyznaczony wcześniej wzór (3), otrzymujemy pochodne składowe funkcji obrazu:

$$\frac{\partial f}{\partial x} = \sin \alpha (z_2 - z_1) \delta(x \sin \alpha - y \cos \alpha + \rho)$$

$$\frac{\partial f}{\partial y} = -\cos \alpha (z_2 - z_1) \delta(x \sin \alpha - y \cos \alpha + \rho)$$

Kierunek gradientu, czyli kierunek, w którym funkcja obrazu wzrasta lub maleje najszybciej, dany jest wzorem:

$$\Theta_{\max} = \begin{pmatrix} \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial x} \end{pmatrix}$$

Lokalną wartość gradientu w danym punkcie obrazu wyznacza się korzystając z normy euklidesowej:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (7)$$

Ze względu na złożoność obliczeniową powyższego wyrażenia wyznaczającego moduł gradientu często stosuje się aproksymację za pomocą wzorów:

$$\|\nabla f\| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right|$$

lub

$$\|\nabla f\| = \max\left(\left| \frac{\partial f}{\partial x} \right|, \left| \frac{\partial f}{\partial y} \right|\right)$$

Moduł gradientu jest operatorem nieliniowym i niezależnym ze względu na rotację.

4.2. Dyskretna realizacja gradientu

Zastosowanie metod gradientowych wykrywania krawędzi polega na dokonaniu splotu obrazu z wybranym operatorem. Operator to para masek umożliwiająca wyznaczenie różniczki obrazu w dwóch prostopadłych kierunkach. Splatając obraz z jedną z macierzy operatora, wyznaczamy tylko jedną składową gradientu tego obrazu. W związku z tym nie zostaną wykryte wszystkie krawędzie. Krawędzie przebiegające prostopadle do orientacji maski nie zostaną wykryte. Aby uzyskać obraz wszystkich krawędzi, bez względu na kierunek ich przebiegu, trzeba zróżniczkować obraz w dwóch prostopadłych kierunkach, a następnie wyznaczyć moduł gradientu ze wzoru (7).

4.3. Podstawowe operatory gradientowe

Najprostszym operatorem gradientowym jest operator Robertsa, którego współczynniki można wyznaczyć bezpośrednio z definicji różniczki:

$$\frac{\partial f}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

Ze względu na to, że w obrazie dyskretnym najmniejszy możliwy przyrost w dowolnym kierunku to jeden piksel ($\Delta x=1$), otrzymujemy:

$$\frac{\partial f}{\partial x} = f(x+1, y) - f(x, y); \quad \frac{\partial f}{\partial y} = f(x, y+1) - f(x, y) \quad (8)$$

W ten sposób otrzymaliśmy maski operatora Roberta (rys. 6 i rys. 7).

-1	1
0	0

-1	0
1	0

Rys. 6. Maski Roberta o orientacji 0° i 90°

Fig. 6. Roberts' masks with orientation 0° and 90°

0	1
-1	0

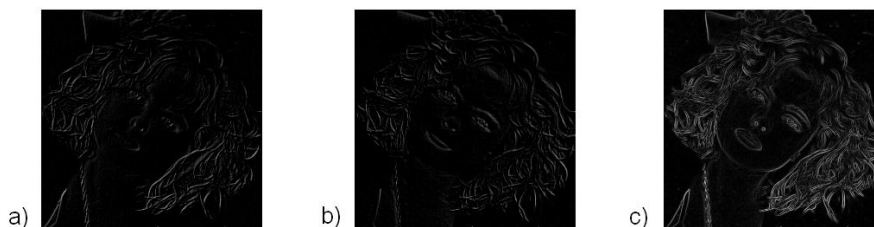
1	0
0	-1

Rys. 7. Maski Roberta o orientacji 45° i 135°

Fig. 7. Roberts' masks with orientation 45° and 135°

Główną zaletą metody Roberta jest szybkość jej działania. W każdym kroku, w celu wyznaczenia wartości piksela na obrazie wyjściowym, brane są pod uwagę tylko cztery piksele obrazu wejściowego. W obliczeniach stosowane jest jedynie dodawanie i odejmowanie, nie występują także żadne parametry algorytmu.

Wadą metody Roberta jest duża wrażliwość na zaszumienie obrazu wejściowego. Wynika to z tego, że algorytm ten stosuje maskę o małych rozmiarach – 2×2. Poza tym algorytm ten daje bardzo słabe odpowiedzi na krawędzie obrazu wejściowego i wykrywa tylko najsilniejsze z nich.



Rys. 8. Zastosowanie operatora Roberta: a) maska pionowa, b) maska pozioma, c) złożenie masek

Fig. 8. Implementation of Roberts' operator: a) vertical mask, b) horizontal mask, c) composition of two masks

Ze względu na to, że operator Roberta jest macierzą 2×2, brak jest w niej wyróżnionego, centralnego piksela. Powoduje to konieczność umownego wyróżniania któregoś z 4 pikseli maski. Dlatego najczęściej we wszelkich operacjach splotu obrazu z pewną maską stosuje się maski o nieparzystej liczbie kolumn i wierszy. Maski Roberta można w łatwy sposób przenieść na maski o rozmiarach 3×3. Powstające w ten sposób maski Prewitta pozwalają różniczkować obraz w różnych kierunkach [1]. Najbardziej popularne są maski Prewitta poziome i pionowe. Rysunek 9. pokazuje struktury tych masek.

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

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

Rys. 9. Pozioma i pionowa maska Prewitta
 Fig. 9. Horizontal and vertical Prewitt's mask

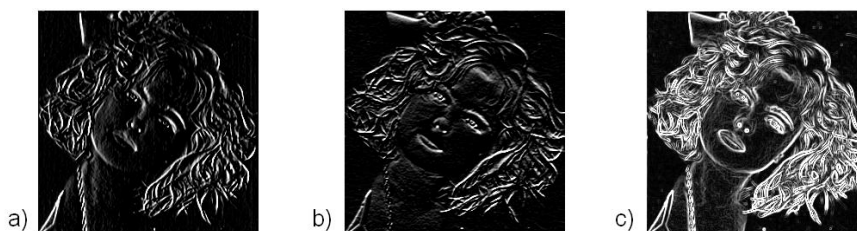
Przy filtracjach gradientowych można także wzmacniać wpływ bezpośrednio najbliższego otoczenia piksela, dla którego wyznaczana jest wartość piksela na obrazie wynikowym. Służą do tego maski Sobela [1]. Mają one postać przedstawioną na rys. 10.

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

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

Rys 10. Pozioma i pionowa maska Sobela
 Fig. 10. Horizontal and vertical Sobel's mask

Działanie operatora Sobela przedstawia rys. 11.



Rys. 11. Zastosowanie operatora Sobela: a) maska pionowa, b) maska pozioma, c) złożenie masek
 Fig. 11. Implementation of Sobel's operator: a) vertical mask, b) horizontal mask, c) composition of two masks

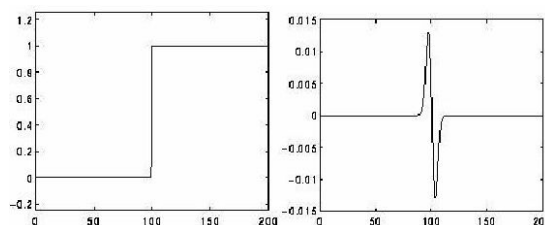
Maski Prewitta i Sobela można swobodnie obracać o 45°. Pozwala to na wyznaczenie gradientu cząstkowego w różnych kierunkach. Poza tym większy rozmiar maski powoduje, że są one mniej podatne na zaszumienie obrazu. Wadą jest konieczność wykonywania większej liczby obliczeń.

Wspólną cechą wszystkich operatorów gradientowych omawianych w tym rozdziale był fakt, że odpowiednie maski operatorów wykazywały określoną asymetrię. Następstwem tego były kierunkowe własności filtrów realizowanych z użyciem tych macierzy. Filtry te podkreślały lub tłumiły pewne cechy obrazu zależnie od tego, pod jakim kątem przebiegały określone krawędzie obiektów na obrazie w stosunku do kierunku asymetrii maski [1]. W celu wyznaczenia wszystkich krawędzi obrazu należało zastosować dwa filtry o różnych (najczęściej prostopadłych) kierunkach asymetrii macierzy. W następnym rozdziale opisane zostaną operatory bezkierunkowe, wykrywające krawędzie bez względu na kierunek ich przebiegu.

5. Metody z użyciem drugiej pochodnej funkcji obrazu

5.1. Obliczanie operatorów drugiej pochodnej w obrazie

Drugą grupą algorytmów wykrywania krawędzi są algorytmy wyznaczające drugą pochodną funkcji obrazu. Jak opisano w poprzednim rozdziale, w miejscu wystąpienia krawędzi, czyli gwałtownego skoku funkcji jasności obrazu, pierwsza pochodna tej funkcji ma ekstremum lokalne. W tym samym miejscu druga pochodna funkcji obrazu ma miejsce zerowe. Wykrywanie krawędzi w przypadku tych algorytmów polega więc na znalezieniu przejść przez zero drugiej pochodnej funkcji obrazu (*zero-crossing*). W odróżnieniu od operatorów gradientowych, operatory przejść przez zero umożliwiają otrzymanie krawędzi o szerokości jednego piksela [2].



Rys. 12. Jednowymiarowa funkcja obrazu zawierającego krawędź skokową oraz druga pochodna tej funkcji

Fig. 12. One dimensional image function for image with step edge and second derivative of this function

Najpopularniejszym operatorem drugiej pochodnej stosowanym do detekcji krawędzi jest operator Laplace'a zwany laplasjanem:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (9)$$

Rzadziej stosowany jest operator drugiej pochodnej w kierunku gradientu:

$$\frac{\partial^2 f}{\partial \vec{n}^2} = \frac{\partial^2 f}{\partial x^2} \cos^2 \alpha + \frac{\partial^2 f}{\partial x \partial y} \sin \alpha \cos \alpha + \frac{\partial^2 f}{\partial y^2} \sin^2 \alpha$$

Wyznaczając drugie pochodne wyrażenia (3), otrzymujemy:

$$\frac{\partial^2 f}{\partial x^2} = \sin^2 \alpha (z_2 - z_1) \delta'(x \sin \alpha - y \cos \alpha + \rho)$$

$$\frac{\partial^2 f}{\partial y^2} = \cos^2 \alpha (z_2 - z_1) \delta'(x \sin \alpha - y \cos \alpha + \rho)$$

Stąd laplasjan funkcji f wynosi:

$$\nabla^2 f = (z_2 - z_1) \delta'(x \sin \alpha - y \cos \alpha + \rho)$$

Laplasjan jest operatorem liniowym i radialnie symetrycznym.

5.2. Dyskretna realizacja laplasjanu

Aby utworzyć dyskretną maskę operatora Laplace'a, dokonujemy różniczkowania funkcji obrazu korzystając ze wzoru (8):

$$\frac{\partial^2 f}{\partial x^2} = f(x+2, y) - 2f(x+1, y) + f(x, y)$$

Następnie możemy dokonać wyśrodkowania, tak aby środkowym pikselem był piksel (x, y) :

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) - 2f(x, y) + f(x-1, y)$$

Postępując analogicznie przy wyznaczaniu drugiej pochodnej w kierunku y , otrzymujemy:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) - 2f(x, y) + f(x, y-1)$$

Stąd otrzymujemy laplasjan:

$$\nabla^2 f = f(x-1, y) + f(x+1, y) - 4f(x, y) + f(x, y-1) + f(x, y+1) \quad (10)$$

Maska operatora odpowiadająca wyrażeniu (10) ma postać przedstawioną na rys. 13.

0	1	0
1	-4	1
0	1	0

Rys. 13. Laplasjan uwzględniający czterosąsiedztwo punktu

Fig. 13. 4 – neighbour Laplace mask

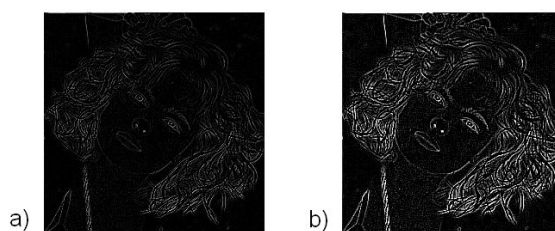
Maska ta uwzględnia jedynie czterosąsiedztwo badanego punktu. Dla polepszenia efektywności detektora zaproponowana została także zmodyfikowana wersja maski Laplace'a, uwzględniająca ośmiospójne sąsiedztwo badanego piksela, przedstawiona na rys. 14.

1	1	1
1	-8	1
1	1	1

Rys. 14. Laplasjan uwzględniający ośmiosąsiedztwo punktu

Fig. 14. 8 – neighbour Laplace mask

Jak widać, maski laplasjanu są symetryczne. Nie ma w nich wyróżnionego kierunku. W związku z tym splot takiej maski z obrazem powoduje wykrycie krawędzi niezależnie od kierunku ich przebiegu. Wadą laplasjanów jest duża wrażliwość na zaszumienie obrazów. Spowodowane to jest tym, że najbardziej wzmacniane są pojedyncze piksele, różniące się w dużym stopniu od swoich czterech lub ośmiu sąsiadów. Zastosowanie obu powyższych masek laplasjanowych do obrazu przedstawia rys. 15.



Rys. 15. Zastosowanie operatora Laplace'a do obrazu: a) maska z 4 sąsiadami, b) maska z 8 sąsiadami

Fig. 15. Implementation of Laplace operator: a) mask with 4 neighbours, b) mask with 8 neighbours

5.3. Algorytm detekcji brzegu z użyciem laplasjano/gaussianu

Algorytm z użyciem laplasjano/gaussianu (*LoG*) jest udoskonaloną metodą detekcji krawędzi przy użyciu filtrów Laplace'a opisanych powyżej. W celu zmniejszenia wrażliwości algorytmu na zaszumienie obraz wejściowy można wygładzić filtrem Gaussa (własności operatora Gaussa były omówione wcześniej). Dwuwymiarowa postać tego operatora dana jest wzorem:

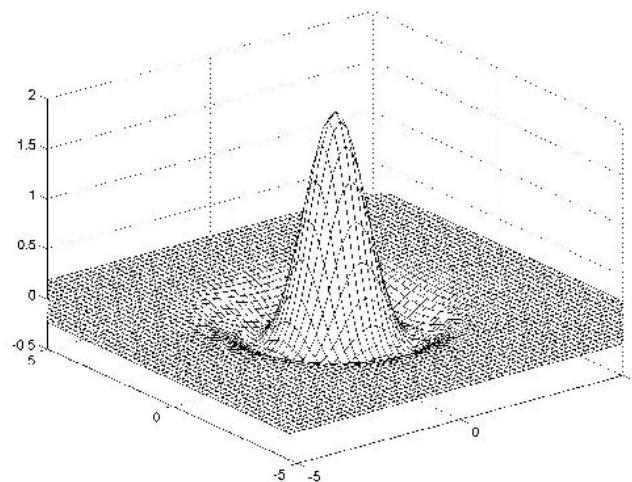
$$G(x, y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (11)$$

Filtr ten jest filtrem uśredniającym, który redukuje zaszumienie obrazu wejściowego. Wygładzony obraz można następnie spleść z operatorem Laplace'a. Ze względu na to, że laplasjan jest operatorem liniowym, możemy nie dokonywać dwukrotnego splatania obrazu (najpierw z operatorem Gaussa, następnie z operatorem Laplace'a). Korzystając z własności operatorów liniowych (6), możemy dokonać splotu operatora Gaussa z operatorem Laplace'a, a następnie powstały operator spleść z obrazem wejściowym [2].

Aby otrzymać operator *LoG* (laplasjan operatora Gaussa), wyznaczamy sumę pochodnych cząstkowych drugiego rzędu po zmiennej x i y z wyrażenia (11):

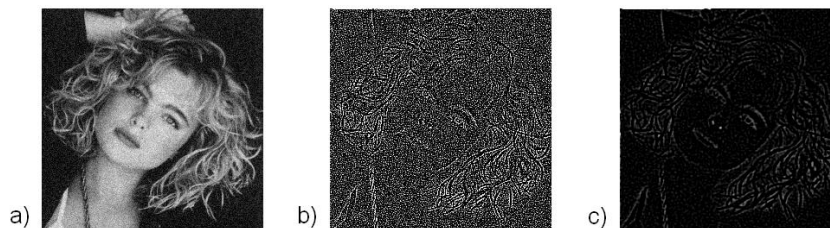
$$LoG = \frac{1}{\pi\sigma^4} \left(\frac{x^2 + y^2}{2\sigma^2} - 1\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (12)$$

Algorytm *LoG* przez stosowanie wygładzania obrazu wejściowego skutecznie wpływa na zmniejszenie zaszumienia obrazu. Stopień uśredniania obrazu zależy od parametru σ w wyrażeniu Gaussa (11). Im większa wartość σ , tym bardziej uśredniony zostanie obraz i większy będzie stopień redukcji szumów. Główną wadą opisywanego algorytmu jest to, że wykryte krawędzie są bardzo słabe. Wynika to z tego, że uśrednianie obrazu filtrem Gaussa rozmywa granice między obszarami o różnej intensywności pikseli. Przejście z jednego obszaru do drugiego nie następuje skokowo, ale płynnie (efekt ten jest tym większy, im większy parametr σ zastosowano w filtrze Gaussa). W związku z tym wykryte krawędzie są na obrazie wyjściowym jeszcze słabiej widoczne niż w przypadku laplasjanów.



Rys. 16. Operator LoG
Fig. 16. LoG operator

Rysunek 17. pokazuje zastosowanie laplasjanu z 8 sąsiadami oraz algorytmu *LoG* do zaszumianego obrazu.



Rys. 17. Zastosowanie laplasjanu: a) zaszumiony obraz wejściowy, b) laplasjan z 8 sąsiadami, c) algorytm LoG dla $\sigma = 1$

Fig. 17. Laplace operator applying: a) source image with noise, b) implementation of Laplace operator with 8 neighbours, c) LoG algorithm for $\sigma = 1$

5.4. Algorytm Difference of Gaussians (*DoG*)

Inną metodą detekcji krawędzi obrazu szukającą przejść przez zero drugiej pochodnej funkcji obrazu jest algorytm Difference of Gaussians (*DoG*). Algorytm ten jest aproksymacją opisanego wcześniej algorytmu *LoG*. *DoG* polega na splocie obrazu wejściowego z dwoma filtrami Gaussa o różnych, odpowiednio dobranych parametrach σ . Wartość piksela na obrazie wyjściowym jest obliczana jako różnica wartości piksela obrazu wejściowego po filtracji pierwszym filtrem Gaussa z parametrem σ_1 i wartości tego samego piksela obrazu wejściowego po filtracji drugim filtrem Gaussa (parametr σ_2)

$$I(x, y) = G_1(x, y) - G_2(x, y) \tag{13}$$

gdzie:

$I(x, y)$ – wartość piksela na obrazie wyjściowym,

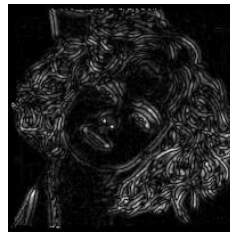
$G_1(x, y)$ – wartość piksela obrazu wejściowego po filtracji pierwszym filtrem Gaussa,

$G_2(x, y)$ – wartość piksela obrazu wejściowego po filtracji drugim filtrem Gaussa.

Wartość piksela $G_1(x,y)$ i $G_2(x,y)$ po filtracji Gaussa wyznacza się ze wzoru (11) podając różne parametry σ . W celu uzyskania najlepszych efektów detekcji zaleca się [2], aby skale σ_1 i σ_2 obu zastosowanych filtrów Gaussa były tak dobrane, by:

$$\frac{\sigma_2}{\sigma_1} \approx 1,6$$

Splot obrazu z dwuwymiarowym filtrem Gaussa (11) jest złożony obliczeniowo. Z tego powodu często stosuje się aproksymację polegającą na splocie obrazu z jednowymiarowym filtrem Gaussa najpierw w kierunku x , potem w kierunku y . Zastosowanie algorytmu *DoG* do obrazu pokazuje rys. 18.



Rys. 18. Algorytm DoG dla $\sigma_1=1$ i $\sigma_2=1,6$

Fig. 18. Implementation of DoG algorithm for $\sigma_1=1$ and $\sigma_2=1,6$

6. Metoda śledzenia brzegu Canny'ego/Deriche'a

6.1. Koncepcja metody

Detektor Canny'ego powstał na bazie wcześniejszych prac opublikowanych przez Marra i Hildretha dotyczących systemów detekcji krawędzi. Canny formułując swoją metodę skoncentrował się na wykrywaniu idealnych krawędzi uskokowych w obecności białego szumu Gaussa. Zawiera ona zarówno elementy podejścia gradientowego, jak i podejścia z zastosowaniem operatora Laplace'a [2].

Punkt wyjścia do badań nad detektorem stanowiły trzy warunki określone przez Canny'ego, które powinny być spełnione przez idealny detektor. Warunki te to:

1. Niski poziom błędów detekcji krawędzi. Prawdopodobieństwo tego, że podczas zaznaczania punktów obrazu jako krawędziowych zaznaczone zostaną punkty nienależące do krawędzi, powinno być jak najmniejsze. Małe powinno być też prawdopodobieństwo ominięcia (niezaznaczenia) punktów krawędziowych.
2. Dobra lokalizacja krawędzi. Przesunięcie między krawędzią wykrytą a rzeczywistą powinno być jak najmniejsze, tak więc punkty zaznaczone przez operator jako punkty krawędzi powinny znajdować się jak najbliżej środka rzeczywistej krawędzi.
3. Pojedyncza odpowiedź na krawędź w obrazie. Dla każdej krawędzi w obrazie powinna istnieć tylko jedna odpowiedź operatora. Ograniczenie to zostało już ujęte w kryterium

pierwszym (niski poziom błędów detekcji) – gdy mamy dwie odpowiedzi na krawędź, jedna z nich na pewno jest błędna.

Dwa pierwsze kryteria nie mogą być jednocześnie w pełni spełnione. Położenie większego nacisku na poprawność detekcji zmniejsza poprawność lokalizacji i odwrotnie.

Poszukiwany detektor, który spełniałby powyższe trzy kryteria, miał być filtrem konwoLucyjnym, który rozmywałby szum i dokonywał detekcji krawędzi [11]. W jednym wymiarze odpowiedź impulsowa filtru f na krawędź L (przy założeniu że filtr ma wartość 0 poza przedziałem $[-W, W]$ jest dana wzorem:

$$H = \int_{-W}^W L(-x)f(x)dx \quad (14)$$

Z matematycznego punktu widzenia [10, 11] trzy kryteria są wyrażone przez następujące równania:

$$SNR = \frac{A \left| \int_{-W}^0 f(x)dx \right|}{n_0 \sqrt{\int_{-W}^W f^2(x)dx}} \quad (15)$$

gdzie:

A – amplituda krawędzi uskokowej,

n_0 – odchylenie standardowe białego szumu Gaussa,

f – odpowiedź impulsowa filtru.

$$Lokalizacja = \frac{A|f'(0)|}{n_0 \sqrt{\int_{-W}^W f'^2(x)dx}} \quad (16)$$

gdzie:

$f(x)$ – pierwsza pochodna odpowiedzi impulsowej filtru.

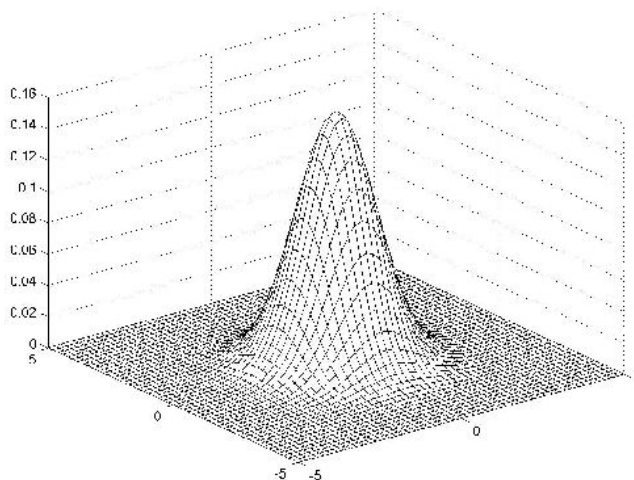
$$x_{zc} = \pi \left(\frac{\int_{-\infty}^{\infty} f'^2(x)dx}{\int_{-\infty}^{\infty} f''^2(x)dx} \right)^{\frac{1}{2}} \quad (17)$$

gdzie:

$f''(x)$ – druga pochodna odpowiedzi impulsowej filtru.

SNR – stosunek sygnału do szumu powinien być jak największy, detektor powinien więc maksymalizować wyrażenie (15). Parametr *Lokalizacja* określa bliskość położenia wyznaczonej krawędzi i krawędzi rzeczywistej – także powinien być jak największy (co oznacza, że odległość krawędzi wyznaczonej od rzeczywistej powinna być minimalna). W związku z

tym wyrażenie (16) także powinno być maksymalizowane. Parametr x_{zc} reprezentuje średnią odległość między przejściami przez zero pierwszej pochodnej funkcji f i określa warunek, aby detektor f nie dawał zbyt dużo odpowiedzi na tę samą krawędź (kilka przejść przez zero w bliskim sąsiedztwie). Uwzględniając powyższe ograniczenia Canny wyznaczył filtr, który maksymalizuje dwa pierwsze warunki i spełnia ograniczenie o wielokrotnych odpowiedziach na pojedynczą krawędź. Ze względu na to, że wynik był zbyt złożony, aby mógł być rozwiązany analitycznie, Canny wyznaczył jego efektywną aproksymację. Jest nim operator pierwszej pochodnej wyrażenia Gaussa. Dwuwymiarowa funkcja Gaussa dana jest wzorem (11), a jej wykres przedstawia rysunek 19.



Rys. 19. Dwuwymiarowy operator Gaussa
Fig. 19. Two dimensional Gauss operator

Dzięki zastosowaniu takiego operatora unika się powstania wielokrotnych odpowiedzi na pojedynczą krawędź, gdyż na wyjściu w sąsiedztwie krawędzi dopuszczalne jest istnienie tylko jednego maksimum lokalnego. Wybór odchylenia standardowego (parametru σ) wyrażenia Gaussa określa skalę algorytmu. Parametr ten wpływa na szerokość filtru oraz stopień wygładzania obrazu. Określa jednocześnie szerokość sąsiedztwa piksela, w którym dozwolone jest tylko pojedyncze maksimum. Oprócz tego, że wybór parametru powinien być uzależniony od tego, jakiej skali obiekty chcemy wykryć, powinien on być też proporcjonalny do wielkości zaszumienia obrazu. Jeżeli wybierzemy zbyt małą wartość parametru, filtr może dawać wielokrotne odpowiedzi na pojedynczą krawędź, a także wykrywać niepożądane krawędzie (tekstury, szum). Jeśli zbyt dużą, zmniejszymy dokładność lokalizacji wykrytej krawędzi [2].

6.2. Algorytm śledzenia brzegu Canny'ego/Deriche'a

Pierwszym krokiem algorytmu Canny'ego jest wygładzenie obrazu wejściowego przez jego splot z filtrem Gaussa (11):

$$I'(x, y) = I(x, y) * G(x, y) \quad (18)$$

gdzie:

$I'(x, y)$ – otrzymany obraz wygładzony,

$I(x, y)$ – obraz wejściowy,

$G(x, y)$ – operator Gaussa.

Splatanie obrazu z dwuwymiarowym filtrem Gaussa jest złożone obliczeniowo. Często stosuje się więc aproksymację tej operacji, polegającą na splataniu obrazu z jednowymiarowym operatorem Gaussa w dwóch prostopadłych kierunkach. Po wykonaniu wygładzenia obrazu następuje wyznaczenie różniczki otrzymanego obrazu w kierunku x oraz y :

$$\nabla_x I'(x, y) = \frac{\partial I'(x, y)}{\partial x}, \quad \nabla_y I'(x, y) = \frac{\partial I'(x, y)}{\partial y} \quad (19)$$

Często w implementacjach zamiast splotu obrazu z wyrażeniem Gaussa, a następnie różniczkowania stosuje się splot z różniczką wyrażenia Gaussa, co jest równoznaczne:

$$\nabla I'(x, y) = \nabla(I(x, y) * G(x, y)) = I(x, y) * \nabla G(x, y) \quad (20)$$

Na podstawie obliczonych pochodnych cząstkowych obrazu wygładzonego $I'(x, y)$ wyznacza się moduł gradientu oraz jego kierunek:

$$M(x, y) = \sqrt{(\nabla_x I'(x, y))^2 + (\nabla_y I'(x, y))^2} \quad (21)$$

$$\Theta(x, y) = \arctan \frac{\nabla_y I'(x, y)}{\nabla_x I'(x, y)} \quad (22)$$

gdzie:

$M(x, y)$ – moduł gradientu,

$\Theta(x, y)$ – kąt nachylenia wektora $M(x, y)$ do osi x układu współrzędnych.

Następnym krokiem jest tłumienie punktów niemaksymalnych (ang. *non-maximal suppression*). W kierunku prostopadłym do krawędzi pozostawia się jako piksel kandydujący tylko ten piksel, w którym moduł gradientu jest maksymalny. Pozostałe piksele są „tłumione” – ustawiane na piksele tła. Operację tę wykonuje się przez testowanie sąsiedztwa 3×3 każdego piksela i porównanie wartości modułu gradientu piksela centralnego z wartościami pikseli sąsiadujących w kierunku gradientu (prostopadle do krawędzi). Jeśli piksel centralny ma wartość maksymalną, zostaje pikselem kandydującym na piksel krawędziowy, jeśli nie przyjmuje wartość tła.

Jak wiadomo, krawędzie mogą być wykryte przez wyznaczenie maksimum gradientu, jak i przejść przez zero drugiej pochodnej obrazu. Algorytm Canny'ego może korzystać także z tej drugiej możliwości podobnie jak laplasjany czy algorytm DoG. W poprzednio opisanych algorytmach druga pochodna była jednak bezkierunkowa – zawierała składową prostopadłą i równoległą do krawędzi. W przypadku algorytmu Canny'ego druga pochodna wyznaczana

jest tylko w kierunku prostopadłym do krawędzi. Piksele, w których wykryto przejście przez zero, zaznaczane są jako kandydujące na punkty krawędziowe [2].

Piksele kandydujące poddawane są następnie progowaniu – za punkty krawędziowe uznawane są tylko te, w których wartość gradientu przewyższa ustalony próg. Stosowanie jednego progu powoduje powstawanie efektu „smug”. Polega on na tym, że wykryta krawędź może być przerywana, nieciągła. Jest to spowodowane wahaniami wyjścia operatora powyżej i poniżej progu. Przypuśćmy, że mamy pojedynczy próg o wartości T_I i istnieje krawędź w obrazie taka, że odpowiedź operatora ma średnią wartość T_I . W takim przypadku będą występowały wahania wyjściowej amplitudy z powodu szumu, nawet jeśli szum jest bardzo mały. Oczekujemy, że wartość krawędzi przekroczy próg średnio połowę razy. To prowadzi do przerwanego konturu na wyjściu. Nieciągłość krawędzi jest bardzo częstym problemem w detektorach krawędzi stosujących progowanie.

Jednym z możliwych rozwiązań tego problemu jest uśrednienie siły krawędzi konturu nad częścią jego długości. Jeżeli średnia ma wartość większą niż próg, cały segment jest zaznaczany, natomiast gdy średnia nie przekracza progu, żadna część konturu nie pojawia się na wyjściu.

W omawianym algorytmie Canny'ego nie są czynione żadne próby presegmentacji konturów. Zamiast tego stosowane jest progowanie z histerezą. Metoda ta polega na ustaleniu dwóch progów. Kandydujący segment krawędzi jest dołączany do wynikowej mapy krawędzi, jeżeli przynajmniej jeden z jego pikseli ma wartość gradientu przekraczającą wyższy próg T_H , a pozostałe piksele nie mają wartości mniejszej niż próg niski T_L [1]:

$$L'(x, y) = \begin{cases} 0, & \text{gdy } L(x, y) \leq T_L \\ s, & \text{gdy } T_L < L(x, y) < T_H \\ 1, & \text{gdy } L(x, y) \geq T_H \end{cases} \quad (23)$$

gdzie:

$L(x, y)$ – wartość piksela w obrazie źródłowym,

$L'(x, y)$ – wartość piksela w obrazie wynikowym $L'(x, y) \in \{0, 1\}$,

T_L – próg niski,

T_H – próg wysoki,

$s=0$ – gdy piksel nie sąsiaduje z pikselem krawędziowym,

$s=1$ – gdy piksel sąsiaduje z pikselem krawędziowym.

Prawdopodobieństwo powstawania „smug” jest poważnie zredukowane, ponieważ kontur musiałby teraz mieć duże wahania wartości: powyżej wysokiego i poniżej niskiego progu. Także prawdopodobieństwo izolowanych punktów fałszywej krawędzi (zaszumienia) jest zredukowane, ponieważ wartość takich punktów musi przekraczać wyższy próg. W zastosowaniach praktycznych stosunek T_H do T_L jest rzędu 2:1 lub 3:1 [10].

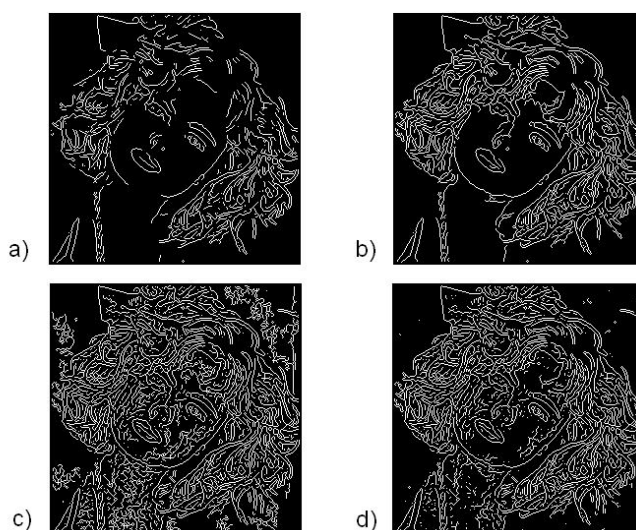
Na rysunku 20. pokazane są efekty działania algorytmu Canny'ego dla różnych wartości parametru σ .



Rys. 20. Algorytm Canny'ego dla różnych wartości parametru σ : a) $\sigma = 0,8$; b) $\sigma = 1,6$; c) $\sigma = 2,4$
 Fig. 20. Canny's algorithm for various values of σ parameter: a) $\sigma = 0,8$; b) $\sigma = 1,6$; c) $\sigma = 2,4$

Na powyższym rysunku widać dokładnie, że ze wzrostem parametru σ algorytm wykrywa krawędzie większej skali, czyli główne kontury obrazu. Szczegóły obrazu dla dużych wartości parametru nie są wykrywane.

Rysunek 21 pokazuje efekt działania progowania z histerezą dla różnych wartości progów.



Rys. 21. Algorytm Canny'ego dla $\sigma=0,8$ i różnych wartości progów: a) $T_H=0,9$; $T_L=0,8$;
 b) $T_H=0,9$; $T_L=0,1$; c) $T_H=0,8$; $T_L=0,1$; d) $T_H=0,8$; $T_L=0,8$;

Fig. 21. Canny's algorithm for $\sigma=0,8$ and various threshold values: a) $T_H=0,9$; $T_L=0,8$;
 b) $T_H=0,9$; $T_L=0,1$; c) $T_H=0,8$; $T_L=0,1$; d) $T_H=0,8$; $T_L=0,8$;

7. Stosowanie wielu detektorów i wielu skal

7.1. Niedostatki algorytmów detekcji brzegów

W komputerowych systemach wizualnych często stajemy przed koniecznością wyboru odpowiedniego detektora krawędzi. Nie jest to zadanie proste, gdyż detektory w większości systemów nie mają wiedzy *a priori* o kontekście przetwarzanego obrazu ani o skali, w jakiej mają działać. Zazwyczaj dokonuje się wyboru jednego detektora, a następnie używa się go do

wykrycia krawędzi we wszystkich obrazach wejściowych. Bardzo trudno jest jednak znaleźć pojedynczy detektor oraz skalę jego działania, która gwarantuje poprawne działanie dla wszystkich krawędzi obrazu. Skala jest najczęściej określana w drodze eksperymentów i stosowana do wszystkich przetwarzanych obrazów. Niestety, w rzeczywistych systemach detektor działający w jednej skali nie będzie działał na tyle poprawnie, by wykryć wszystkie krawędzie obrazu. Dlatego w takich wypadkach często stosuje się kilka różnych detektorów, o różnych własnościach i skalach działania. W takim przypadku jednym z podstawowych problemów jest wybór odpowiednich detektorów i skal. Innym rozwiązaniem tego problemu jest zastosowanie jednego detektora działającego w kilku skalach. Po zastosowaniu małej skali działania algorytmu wykryte krawędzie mogą zawierać wiele krawędzi fałszywych (ze względu na to, że algorytmy małej skali są bardzo wrażliwe na zaszumienie). Inną cechą algorytmów małej skali jest to, że otrzymana mapa krawędzi zawiera szczegóły obrazu (wykryte są krawędzie małej mocy, czyli odpowiadające małym zmianom intensywności obrazu). Przy zastosowaniu algorytmu dużej skali otrzymujemy tylko kontury głównych obiektów obrazu (brak szczegółów), ale wykryte krawędzie mogą mieć duży błąd lokalizacji [9].

7.2. Wieloskalowa detekcja i jej uwarunkowania

Gdy zastosujemy wiele skal działania algorytmu, otrzymamy kilka map krawędzi (każda otrzymana z detektora o pewnej określonej skali). Otrzymane mapy krawędzi trzeba następnie połączyć w jedną wynikową mapę krawędzi. W opisywanym podejściu pojawiają się dwa problemy:

- Jak dobrać ilość różnych skal, w których powinien działać detektor oraz jakie wartości powinny mieć te skale.
- Jak dokonać połączenia map odpowiadających różnym skalom w jedną mapę wynikową.

W omawianym podejściu detektory działają więc w różnych skalach: $s_1, s_1 + \Delta s, s_1 + 2\Delta s, \dots, s_2$. Jeśli Δs jest ujemna, algorytm wykrywa najpierw ogólne zarysy sceny, a w następnych krokach coraz bardziej szczegółowe informacje (podejście *kontur – szczegół*). Gdy Δs jest dodatnia, najpierw wykrywamy szczegóły, potem ogólną scenę (podejście *szczegół – kontur*). Wybór $s_1, \Delta s, s_2$ zależy od charakterystyk obrazu i własności detektora. Δs powinno być jednak na tyle małe, że różnice między kolejnymi mapami krawędzi (otrzymanymi w kolejnych krokach) będą niewielkie i łatwiejsze będzie przeprowadzenie operacji łączenia tych map w jedną mapę wynikową. Są także próby zastosowania Δs zmieniającego się dynamicznie w trakcie działania algorytmu [9].

Proces łączenia map krawędzi w mapę wynikową polega na zaetykietowaniu jako jednej krawędzi wielu krawędzi otrzymanych w różnych skalach, a pochodzących od jednej krawędzi rzeczywistej. Odpowiadające sobie krawędzie są łączone w jeden obraz, poczym doda-

wane są krawędzie, które nie są do tej pory dopasowane (nie występowały we wcześniej rozpatrywanych mapach). Łączenie obrazów jest procesem łatwiejszym jeśli detektor ma odpowiednią własność – ze wzrostem skali ilość wykrytych krawędzi maleje. Oznacza to, że w podejściu *szczegół – kontur* jeśli pewna krawędź nie jest już wykrywana w pewnej skali (choć była wykrywana przy zastosowaniu mniejszej skali), to na pewno nie pojawi się już na mapie otrzymanej dla większej skali.

Przykładem praktycznego zastosowania opisanego podejścia z wykorzystaniem wielu skal działania detektorów jest *feature synthesis* w algorytmie zaproponowanym przez Canny'ego (jest to algorytm *szczegół – kontur*). W algorytmie tym dzięki zastosowaniu różnych wartości parametru σ w operatorze Gaussa Canny otrzymał zbiór map krawędzi o różnej skali. Może istnieć wiele skal, w których dana krawędź jest wykrywana – w takim przypadku powinien zostać wybrany operator o najmniejszej skali, gdyż daje on najlepszą lokalizację. Krawędzie niewykryte przez operator najmniejszej skali mogą być wykryte w następnych krokach przez detektor o większej skali. Gdy tylko detektor z najmniejszą skalą wykryje krawędź, powinna być ona zaznaczona. Gdy detektory większej skali wykryją także tę krawędź (mimo że nie będą się one pokrywały, gdyż ze wzrostem skali rośnie błąd lokalizacji), Canny proponuje zastosowanie *feature synthesis*. Metoda ta polega na tym, że zaczynamy od zaznaczenia wszystkich krawędzi z detektora o najmniejszej skali. Korzystając z otrzymanych krawędzi staramy się przewidzieć wyjście detektora o większej skali (gdyby na wejście podać mu tylko krawędzie otrzymane w pierwszym kroku). Następnie porównujemy rzeczywiste wyjście detektora większej skali z przewidywanym przez nas wyjściem. Zaznaczamy dodatkowe krawędzie tylko wtedy, gdy detektor daje znacznie mocniejszą odpowiedź niż ta, którą przewidywaliśmy. Ta procedura może być realizowana iteracyjnie z zastosowaniem detektorów coraz większej skali. W pierwszym kroku zaznaczamy krawędzie wykryte przez detektor o najmniejszej skali, w drugim dodajemy krawędzie wykryte przez detektor większej skali, w kolejnym przez detektor o jeszcze większej skali itd. W ten sposób otrzymujemy narastającą, kumulacyjną mapę krawędzi (w każdej skali dodawane są krawędzie, które nie były zaznaczone w mniejszych skalach). Takie podejście jest bardziej efektywne niż zwykle nakładanie na siebie wyjść detektorów różnej skali, gdyż w tym drugim przypadku mogą wystąpić różnice w lokalizacji tych samych krawędzi wykrytych przez różne operatory [10, 3].

8. Zakończenie

Proces detekcji krawędzi jest realizowany w trzech etapach: wygładzanie obrazu, różniczkowanie i etykietowanie krawędzi. Dwa pierwsze kroki, w przypadku operatorów linio-

wych, mogą być wykonywane w dowolnej kolejności. Wygładzanie ma na celu redukcję zaszumienia obrazu. Dzięki zastosowaniu na tym etapie operatorów dokonujących regularyzacji różniczkowania (np. operator Gaussa) możemy dokonać wyboru skali działania algorytmu detekcji. Przy wyborze małej skali otrzymujemy szczegóły obrazu, ale także wiele fałszywych krawędzi. W dużej skali otrzymujemy jedynie zarysy głównych obiektów sceny, jednak wykryte krawędzie mają duży błąd lokalizacji. Ze względu na to, że trudno jest dokonać wyboru właściwej skali działania algorytmu detekcji, stosuje się podejście z wieloma skalami. Jeden obraz podawany jest na wejście kilku detektorów o różnych skalach. Następnie otrzymane w ten sposób mapy krawędzi są łączone w jedną mapę wynikową.

W artykule omówiono także podstawowe i najczęściej stosowane metody detekcji krawędzi obrazu. Algorytmy gradientowe, których charakterystyczną cechą jest kierunkowość, wyznaczają pierwszą pochodną funkcji obrazu i szukają ekstremów lokalnych tej pochodnej. W miejscu ich wystąpienia na obrazie znajduje się krawędź. Najprostszym operatorem gradientowym jest operator Roberts'a. Ze względu na brak centralnego piksela i dużą wrażliwość na zaszumienie tego operatora częściej stosuje się operatory Prewitta lub Sobela. W odróżnieniu od gradientowych, algorytmy przejść przez zero (laplasjany, *LoG*) są bezkierunkowe. Wyznaczają one drugą pochodną funkcji obrazu i w celu wykrycia krawędzi szukają miejsc zerowych obliczonej pochodnej. Wadą tych metod jest bardzo duża wrażliwość na zaszumienie obrazu oraz oczywiście to, że nie możemy otrzymać obrazu krawędzi o pewnym wyróżnionym kierunku. W przypadku algorytmu *LoG*, którego operator powstaje przez wyznaczenie laplasjanu wyrażenia Gaussa, pojawia się problem odpowiedniego doboru parametru σ . Duża wartość zapewnia redukcję szumów, ale jednocześnie powoduje, że algorytm działa w dużej skali, czyli wykrywa zarysy głównych obiektów obrazu pomijając szczegóły. Chęć wykrycia elementów małej skali – szczegółów obrazu, tekstur, cieni zmusza do zmniejszenia wartości parametru σ , co z kolei wpływa na zmniejszenie poziomu redukcji zaszumienia. Głównym problemem jest znalezienie kompromisu między tymi dwoma właściwościami parametru σ .

BIBLIOGRAFIA

1. Tadeusiewicz R., Korohoda P.: Komputerowa analiza i przetwarzanie obrazów. Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków 1997.
2. Bovik A. C.: Handbook of Image and Video Processing. London 2001.
3. Samko M.: Dydaktyczny system detekcji krawędzi obrazu – praca magisterska. Politechnika Gdańska, 2002.
4. Strony internetowe: <http://plum.ia.polsl.gliwice.pl/~dip/hipr/roberts.html>, 06.2005.

5. Strony internetowe: <http://plum.ia.polsl.gliwice.pl/~dip/hipr/sobel.html>, 06.2005.
6. Strony internetowe: <http://plum.ia.polsl.gliwice.pl/~dip/hipr/log.html>, 06.2005.
7. Ritter G., Wilson J. : Handbook of Computer Vision Algorithms in Image Algebra. London 1996.
8. Strony internetowe: <http://marathon.csee.usf.edu/edge>, 12.2008.
9. Ziou D., Tabbone S.: Edge detection techniques – an overview. Int. J. Pattern Recognit. Image Anal. v. 8, i. 4. s. 537÷559.
10. Canny J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8, 1986.
11. Advanced Edge Detection Technique: Techniques in Computational Vision: <http://www.cpsc.ucalgary.ca/Research/vision/501/edgedetect.pdf>, 03.2009.
12. Heath M., Sarkar S., Sanocki T., Bowyer K.: Comparison of Edge Detectors. Computer Vision and Image Understanding, Vol. 69, No. 1, 1998.

Recenzent: Dr inż. Adam Świtoński

Wpłynęło do Redakcji 24 lutego 2009 r.

Abstract

The aim of the article was to present various edge detection methods in monochromatic digital images. It described gradient detectors, second derivative detectors and the Canny's algorithm. The detectors and the algorithm were described on the basis of the source image: a digital black and white picture of a woman's face.

Firstly, main stages of edges detection process, their characteristics and the main conditions that the processes must fulfill were presented. Detection of lines, edges and crossings were described. Secondly, different kinds of detectors were enumerated and the processes of smoothing, differentiating and labeling the image were presented. Then, the article showed two basic groups of edges detection methods: gradient and second derivative methods.

Gradient methods calculate the first derivative of the image function. These algorithms are directional algorithms, which means that they allow detecting edges going in a given direction. In order to obtain all the edges it is necessary to use the algorithm in two perpendicular directions. The main gradient operators include Roberts', Prewitt's and Sobel's operators.

Second derivative algorithms, which are non-directional algorithms, result in a set of edges independent from their direction. The algorithm is based on the second derivative of

After discussing gradient and zero-crossing methods, the article described Canny/DeRiche method of border tracing. The algorithm includes smoothing the input image by its convolution with the Gaussian filter, defining the module and direction of the gradient followed by non-maximal suppression and thresholding with hysteresis. The algorithm allows to detect main contours of the image by increasing the threshold value.

Finally, the article showed deficiencies of border detection methods and described multiscale detection, its conditions and main problems: choosing the number and values of scales and joining maps of various scales into one.

Adres

Marek SAMKO: Uniwersytet Warmińsko – Mazurski, Wydział Matematyki i Informatyki,
ul. Żołnierska 14, 10-561 Olsztyn, Polska, samko@matman.uwm.edu.pl