

Tomasz PODESZWA, Krzysztof CYRAN  
Politechnika Śląska, Instytut Informatyki

## DETEKCJA KRAWĘDZI - PRZEGLĄD ALGORYTMÓW

**Streszczenie.** W artykule przedstawiono przegląd algorytmów wykorzystywanych podczas detekcji krawędzi w obrazach monochromatycznych (o 256 stopniach szarości), ze szczególnym uwzględnieniem metod opartych na badaniu zmian jasności w ortogonalnych osiach rastra obrazu prostych masek splotu. Przedstawiono wyniki ich działania dla przykładowego obrazu, jak również idee bardziej zaawansowanych metod.

## EDGE DETECTION - ALGORITHMS OVERVIEW

**Summary.** The article presents overview of edge detection algorithms applied to monochromatic (grayscale) images. This methods based mainly on examine pixels brightness in two orthogonal directions in an image using simple convolution masks. Examples of pictures after use this masks are included in this article.

### 1. Wprowadzenie

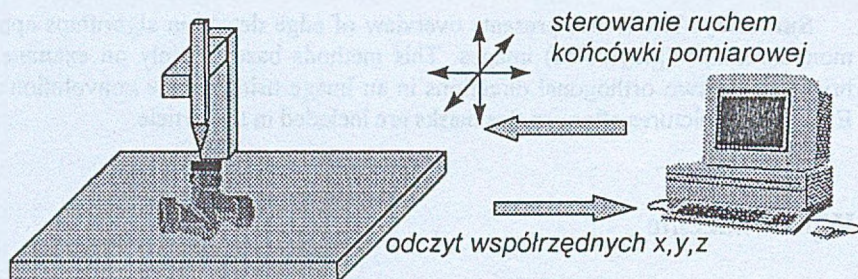
Detekcja (wykrywanie) krawędzi jest jedną z najczęściej wykorzystywanych w analizie obrazów operacji, pozwalającą na jakościową ocenę ich zawartości (umożliwiająca w kolejnych etapach określenie rodzaju i ilości obiektów występujących na obrazie).

Typowym przykładem wykorzystania tego typu algorytmów może być układ pomiarowy złożony z kamery, przetwornika analogowo-cyfrowego oraz komputera z odpowiednim oprogramowaniem służącym analizie otrzymanych obrazów. Jednym z zastosowań takiego zestawu (uzupełnionego o układ sterowania położeniem kamery w kierunkach X, Y i Z, aby uzyskać większy obszar możliwy do przemierzenia - rys. 1.) jest jeden z ostatnich etapów procesu produkcji, czyli kontrola jakości mająca na celu sprawdzenie, czy dany wyrób spełnia

wymagania założone w projekcie oraz normy, co w przypadku elementów mechanicznych wiąże się głównie z pomiarem ich wymiarów geometrycznych oraz porównaniem, czy wielkości charakterystyczne zostały zachowane.

W przypadku pomiaru ręcznego operator ręcznie wymusza ruch czujnika dotykowego, który po napotkaniu przeszkody zatrzymuje silniki. Następnie operator odczytuje położenie (współrzędne  $x,y,z$ ). Po wykonaniu co najmniej kilku takich pomiarów następuje przeliczenie wartości w celu porównania otrzymanych wymiarów elementów ze wzorcem. W przypadku zastosowania komputera do sterowania ruchem czujnika oraz jako urządzenia przeliczającego zmniejsza się co prawda czas pomiaru, lecz ilość wymaganych ruchów pozostaje podobna. Zmniejszenie ilości ruchów jest możliwe po zastosowaniu zamiast czujnika dotykowego układu złożonego z kamery pozwalającej objąć większy obszar mierzonego przedmiotu. Wymaga to jednak zastosowania bardziej skomplikowanych metod przeliczania uzyskanych danych, w szczególności doboru odpowiednich algorytmów analizy otrzymanych obrazów ze względu na dokładność odwzorowania krawędzi, zdolność do rozróżniania niewielkich szczegółów oraz odporność na zakłócenia. Ważna jest również złożoność obliczeniowa poszczególnych metod, która najczęściej wzrasta wraz z polepszeniem wcześniej wymienionych parametrów.

W artykule przedstawione zostaną różne metody używane do detekcji krawędzi wraz z przykładowymi wynikami ich zastosowania.



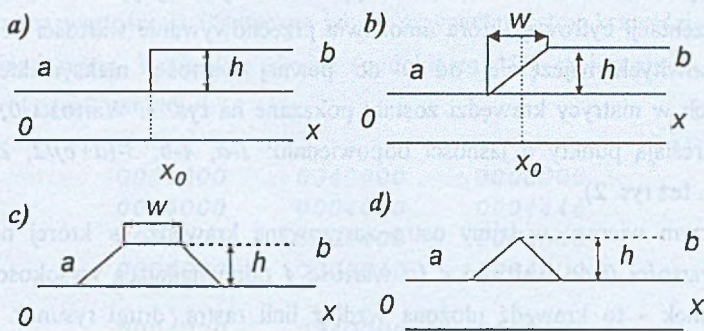
Rys. 1. Schemat systemu pomiarowego  
Fig. 1. Measure system scheme

## 2. Modele krawędzi, linii i punktu.

Wszystkie algorytmy detekcji krawędzi (oraz linii i punktu) są oparte na badaniu zmian amplitudy jasności w obrazie źródłowym, które najczęściej wiążą się przebiegiem krawędzi w przedstawianym na obrazie obiekcie.



Na rys. 2. przedstawiono typowe (idealne) modele krawędzi i linii. We wszystkich tych modelach przyjęto, że poziom jasności zmienia się od wartości  $a$  do  $b$ . W zależności od przebiegu tych zmian, wyróżniamy kilka typowych przypadków. W pierwszym z nich (rys. 2a) widzimy idealną krawędź o wysokości  $h$ , w której następuje skokowa zmiana jasności od  $a$  do



Rys. 2. Modele krawędzi i linii  
 Fig. 2. Edge and line models

$b$  i chociaż w warunkach rzeczywistych wydawałoby się, że przypadek ten jest częsty, to jednak po uwzględnieniu procesu rasteryzacji i dyskretyzacji oraz innych zjawisk fizycznych (np. związanych z optyką kamery) następuje rozmycie ostrości krawędzi, czego wynikiem jest otrzymanie krawędzi, której zmiana jasności następuje w obszarze o szerokości  $w$  (rys. 2b).

Trzeci rysunek (rys. 2c) przedstawia dwie krawędzie znajdujące się dostatecznie blisko siebie (odległość równa w przybliżeniu wielkości rastra), tworzące linię o szerokości  $w$  i wysokości  $h$ , w której przejście pomiędzy wartościami  $a$  i  $b$  następuje w sposób płynny. Kolejny rysunek (rys. 2d) to przykład linii, której szerokość  $w=0$ , co może być rezultatem wąskiej linii lub np. wybrania rastra o dość dużej szerokości. W obu tych przypadkach uśredniające działanie przetwornika A/C spowoduje zmniejszenie wysokości linii, co w obrazie wynikowym uwidoczni się jako liniowe (szerokości jednego lub dwóch punktów) rozjaśnienie o wartości zależnej od stosunku szerokości linii w stosunku do szerokości rastra oraz położenia linii w stosunku do rastra.

Jeżeli dodatkowo zmiana jasności obrazu zachodzi w obu ortogonalnych kierunkach ( $x$  oraz  $y$ ) wokół badanego punktu (zgodnie z rys 2c lub 2d) tworząc lokalne ekstremum, to otrzymujemy model punktu o wysokości  $h$ .

### 3. Reprezentacja po dyskretyzacji

Obraz przechodzący przez obiektyw po dojściu do matrycy CCD ulega w niej rasteryzacji, a następnie (w karcie akwizycji wyposażonej w odpowiednio szybki przetwornik) kwantyzacji. Pierwsze jest spowodowane skończoną ilością punktów tej matrycy, natomiast drugie wynika z samej reprezentacji cyfrowej, która umożliwia przechowywanie wartości ze skończonego zakresu (całkowitych najczęściej od 0 do pewnej wartości maksymalnej). Przykłady przetworzonych w matrycy krawędzi zostały pokazane na rys. 3. Wartości 0, 1, 2, 3, 4 są umowne i określają punkty o jasności odpowiednio:  $1-a$ ,  $4-b$ ,  $3-(a+b)/2$ ,  $2-(3a+b)/4$ ,  $3-(a+3b)/4$  (por. też rys. 2).

W pierwszym wierszu widzimy ostro zarysowaną krawędź, w której następuje skok jasności od wartości 0 do wartości 4 (o wartość 4 odpowiadającą wysokości  $h$  z rys. 2). Pierwszy rysunek - to krawędź ułożona wzdłuż linii rastra, drugi rysunek - to krawędź ułożona pod kątem  $45^\circ$  do rastra, natomiast ostatni przypadek - to przykład narożnika. Jest to przypadek stosunkowo rzadki (jeśli chodzi o obrazy przetworzone z sygnału wizyjnego w karcie akwizycji, w których to następuje rozmycie krawędzi, jak to zostało przedstawione w poprzednim rozdziale) przydatny jednak do analizy algorytmów pod względem dokładności określania położenia krawędzi.

Drugi wiersz przedstawia tę samą krawędź (o szerokości  $w$  równej dwóm punktom), jeżeli znalazła się ona pomiędzy liniami rastra. Porównując otrzymane wartości z poprzednim przypadkiem, należy zauważyć rozmycie krawędzi, którego rezultatem jest pojawienie się

0004444	0044444	000000
0004444	0004444	000444
0004444	0000444	000444
0004444	0000044	000444
0002444	0024444	000000
0002444	0002444	000222
0002444	0000244	000244
0002444	0000024	000244
0002444	0134444	000000
0002444	0013444	000122
0002444	0001344	000244
0002444	0000134	000244

Rys. 3. Reprezentacja krawędzi po rasteryzacji

Fig. 3. Discrete edge model



próbek o wartości pośredniej  $(a+b)/2$  (wartość 2). Jak zostanie to przedstawione w dalszej części niniejszej pracy, powoduje to pogorszenie dokładności pomiaru oraz konieczność stosowania algorytmów operujących na większej ilości sąsiednich punktów obrazu.

W trzecim wierszu przedstawiono dane otrzymane po rasteryzacji dla krawędzi o szerokości równej wielkości rastra. Dla krawędzi o orientacji zgodnej z rastrem (rysunek górny) otrzymane wartości są identyczne jak w przypadku ostrej krawędzi znajdującej się pomiędzy liniami rastra. Krawędzie skośne (pod kątem  $45^\circ$ ) są jeszcze bardziej rozmyte (obszar o szerokości 2 punktów).

0004000	0040000	0000000
0004000	0004000	0004444
0004000	0000400	0004000
0004000	0000040	0004000
0024200	0242000	0000000
0024200	0024200	0222222
0024200	0002420	0244444
0024200	0000242	0242222
		0242000
0002200	0121000	0000000
0002200	0012100	0012222
0002200	0001210	0023222
0002200	0000121	0022000

Rys. 4. Reprezentacja linii po rasteryzacji  
 Fig. 4. Discrete line model

Podobnie jak w przypadku krawędzi, również w przypadku linii proces rasteryzacji powoduje utratę części informacji. Na rys. 4. widzimy różne przypadki położenia linii o szerokości jednego punktu i skokowej (pierwsza kolumna) oraz płynnej (druga i trzecia kolumna) zmianie jasności.

Pierwszy wiersz przedstawia przypadek idealny: zarówno przejście pomiędzy wartościami  $a$  (wartość 0 na rysunku) i  $b$  (wartość 4 na rysunku) jest wyraźnie zaznaczone (skokowe), jak również położenie względem rastra było zgodne z kierunkiem rastra przetwornika.

Dla przypadku częściej spotykanego w rzeczywistości wynik przetworzenia jest przedstawiony w drugim wierszu (dla położenia linii dokładnie w linii rastra) oraz w wierszu trzecim (linia przesunięta o połowę szerokości rastra). Istotne dla dalszego etapu (czyli detekcji linii) jest to, że linia straciła na ostrości (nastąpiło poszerzenie jej obrazu) oraz jasności (obniżenie różnicy pomiędzy wartością maksymalną a minimalną). Warto zauważyć, że dla linii o szerokości jednego punktu, znajdującej się w narożniku, otrzymuje się optyczne zaokrąglenie utrudniające dokładną detekcję punktu narożnego.

0000000	0000000	0000000
0000000	0022200	0000000
0004000	0024200	0001100
0000000	0022200	0001100
0000000	0000000	0000000

Rys. 5. Reprezentacja punktu po rasteryzacji  
Fig. 5. Discrete pixel spot model

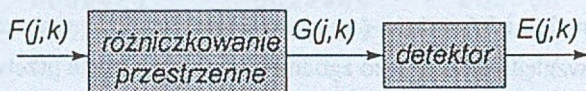
Ostatnim (i najtrudniejszym) przypadkiem jest detekcja pojedynczego punktu. Na rys. 5. widzimy przypadek idealny (rysunek z lewej), w którym punkt ma wysokość  $h=b-a$  i znajduje się dokładnie w siatce rastra oraz dwa przypadki punktu, w którym przejścia są liniowe z różnym ułożeniem względem rastra.

Dla punktu położonego w rastrze (rysunek środkowy) otrzymany obraz pozwala jeszcze na prawidłowe rozpoznanie położenia punktu, chociaż może być to utrudnione z powodu rozmycia przejścia od wartości  $a$  do  $b$ . Jeżeli dodatkowo taki punkt nie znajduje się dokładnie w rastrze przetwornika, następuje tak znaczne rozmycie punktu, że jego detekcja może się okazać praktycznie niemożliwa (rysunek z lewej).

#### 4. Wyznaczanie gradientu w ortogonalnych osiach rastra

Obraz uzyskany z kamery i przetworzony na obraz rastrowy podlega w następnym kroku przetworzeniu na obraz różnicowy, który z kolei poddawany jest progowaniu (rys. 6.) mającemu na celu odfiltrowanie zakłóceń (szumów) znajdujących się w każdym rzeczywistym sygnale.

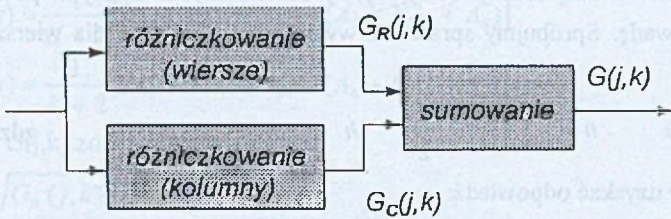
Założmy, że obraz złożony jest z  $j$  wierszy i  $k$  kolumn, oznaczmy przez  $F(j,k)$  obraz pierwotny (po rasteryzacji),  $G(j,k)$  - obraz po różniczkowaniu (wyznaczeniu gradientu) oraz przez  $E(j,k)$  obraz po przetworzeniu w detektorze progowym.



Rys. 6. Etapy wykrywania krawędzi, linii i punktów  
Fig. 6. Edge, line and pixel spot detection steps

Pierwszym z etapów detekcji krawędzi jest dokonanie na oryginalnym obrazie  $F(j,k)$  operacji wykrycia zmian w jasności sąsiadujących punktów. Wykorzystuje się w tym celu różniczkowanie w dwóch prostopadłych osiach (np.  $x$  i  $y$ ), których wypadkowe  $G_R(j,k)$  i  $G_C(j,k)$  są następnie sumowane (rys. 7.).





Rys. 7. Schemat realizacji różniczkowania przestrzennego  
 Fig. 7. Orthogonal gradient generation

Dla krawędzi położonej pod kątem  $\theta$  do linii rastrowi otrzymujemy wzór ([1], s. 106):

$$G(x, y) = \frac{\partial F(x, y)}{\partial x} \cos(\theta) + \frac{\partial F(x, y)}{\partial y} \sin(\theta) \quad (1)$$

Można to przedstawić schematycznie (jak na rys. 7.) przyjmując:

$$G_R = \frac{\partial F(x, y)}{\partial x}, G_C = \frac{\partial F(x, y)}{\partial y} \quad (2)$$

i otrzymując uproszczony wzór:

$$G(j, k) = \sqrt{G_R(j, k)^2 + G_C(j, k)^2} \quad (3)$$

Korzystając z tego wzoru uzyskujemy dokładnie taką samą jasność krawędzi prostopadłych do rastrowi, jak również krawędzi biegnących pod kątem. Ponieważ jednak wykorzystuje się w nim operacje o dość dużej złożoności arytmetycznej (potęgowanie oraz pierwiastkowanie), częściej używane jest wyrażenie:

$$G(j, k) = |G_R(j, k)| + |G_C(j, k)| \quad (4)$$

w którym normę kartezjańską zastąpiono normą „manhattan” wykorzystującą bardzo prostą w realizacji (przez co szybko) funkcję moduł liczby (wartość bezwzględna). Niestety, dla tak zdefiniowanej operacji sumowania krawędzie pionowe i poziome (zgodne z rastrem) są o około 30% słabsze od identycznych krawędzi położonych pod kątem 45% do rastrowi. Jak zostanie to przedstawione w dalszej części, można temu częściowo zaradzić dobierając odpowiednio współczynniki w algorytmach wyznaczających  $G_R(j, k)$  oraz  $G_C(j, k)$ .

#### 4.1. Przykłady funkcji wyznaczających gradienty $G_R$ i $G_C$

Najprostszą funkcją wyznaczającą gradienty  $G_R$  i  $G_C$  jest różnica dwóch sąsiadujących ze sobą punktów:

$$G_R(j, k) = F(j, k) - F(j, k-1) \text{ dla wiersza oraz}$$

$$G_C(j, k) = F(j, k) - F(j-1, k) \text{ dla kolumny.}$$

Dla tak zdefiniowanej funkcji otrzymujemy bardzo szybki algorytm posiadający jednak dość poważną wadę. Spróbujmy sprawdzić wynik różniczkowania dla wiersza składającego się z ciągu

$$0 \quad 0 \quad 0 \quad 0 \quad \frac{h}{2} \quad h \quad h \quad h \quad h \quad \text{gdzie } h=b-a$$

Powinniśmy uzyskać odpowiedź:

$$0 \quad 0 \quad 0 \quad 0 \quad h \quad 0 \quad 0 \quad 0 \quad 0$$

Jak łatwo obliczyć,  $|G_R(j,k)|$  będzie miał wartości:

$$0 \quad 0 \quad 0 \quad 0 \quad \frac{h}{2} \quad \frac{h}{2} \quad 0 \quad 0 \quad 0$$

Można zauważyć, że dla tak dobranej funkcji nie da się jednoznacznie zlokalizować położenia punktu, gdyż następuje rozmycie oraz przesunięcie szczytu sygnału. Aby temu zapobiec, należy zastosować funkcje, które są np. różnicą punktów sąsiadujących z punktem przejścia. Przyjmując funkcję różniczkującą

$$G_R(j,k) = F(j,k+1) - F(j,k-1) \text{ dla wiersza oraz}$$

$$G_C(j,k) = F(j-1,k) - F(j+1,k) \text{ dla kolumny.}$$

Dla krawędzi

$$0 \quad 0 \quad 0 \quad 0 \quad \frac{h}{2} \quad h \quad h \quad h \quad h$$

otrzymujemy odpowiedź:

$$0 \quad 0 \quad 0 \quad \frac{h}{2} \quad h \quad \frac{h}{2} \quad 0 \quad 0 \quad 0$$

Jakkolwiek wynik, który otrzymaliśmy, jest zadowalający w przypadku krawędzi, to jednak w przypadku linii w obrazie wynikowym otrzymujemy (oprócz krawędzi) wartości będące m.in. wynikiem fluktuacji występujących w sygnale wizyjnym (szumy elementów analogowych).



Rys. 8. Schemat numerowania komórek

Fig. 8. Cell numbering convention

Można temu zapobiec poprzez różniczkowanie w jednym kierunku z równoczesnym uśrednianiem w kierunku prostopadłym. Jeśli oznaczymy komórki sąsiadujące w sposób przedstawiony na rys. 8, to funkcje  $G_R$  i  $G_C$  można przedstawić w postaci:



$$G_R(j, k) = \frac{1}{K+2} [(A_2 + KA_3 + A_4) - (A_0 + KA_7 + A_6)]$$

$$G_C(j, k) = \frac{1}{K+2} [(A_0 + KA_1 + A_2) - (A_6 + KA_5 + A_4)]$$
(5)

Dla funkcji  $G(j, k)$  zdefiniowanej jako

$$G(j, k) = \sqrt{G_R(j, k)^2 + G_C(j, k)^2}$$

oraz  $K=1$  (podanej przez Prewitta [1]) otrzymujemy najprostszą postać funkcji różniczkująco-uśredniającej. Dla  $K=2$  (Sobel, [2] s.271) otrzymujemy wyrównanie wag punktów położonych  $K = \sqrt{2}$  na liniach środkowych (np.  $A_3, A_7$ ) z wagami punktów położonych poza nimi (np.  $A_0, A_2, A_4, A_6$ ), co powoduje wyrównanie wpływu każdego z punktów znajdujących się w komórkach  $A_{\sigma \div A_7}$  na wartość  $G(j, k)$ . W przypadku gdy zależy nam na identycznej odpowiedzi układu zarówno dla krawędzi pionowych i poziomych oraz ułożonych po przekątnej, należy przyjąć (Frei i Chen, [3]).

Dla krawędzi o liniowym przejściu szerokości jednego punktu i wysokości  $h$  (rys. 2b oraz rys. 3. wiersz drugi) dla wszystkich tych masek otrzymujemy:

dla krawędzi znajdującej się w linii rastra (por. rys. 3. z lewej) kolejno:

$$0 \quad 0 \quad \frac{h}{2} \quad h \quad \frac{h}{2} \quad 0 \quad 0$$

dla krawędzi znajdującej się skośnie (kąt  $45^\circ$ ) do linii rastra (por. rys. 3. w środku):

$$0 \quad \frac{h}{\sqrt{2}(2+K)} \quad \frac{h}{2} \quad \frac{\sqrt{2}(1+K)h}{2+K} \quad \frac{h}{2} \quad \frac{h}{\sqrt{2}(2+K)} \quad 0$$

Powyższe wartości otrzymujemy po zastosowaniu równań (5) dla kolejnych punktów linii rastra (np. poziomych) przecinających krawędź.

Przedstawione powyżej algorytmy działają w pewnym otoczeniu przejścia i są liniową kombinacją jasności sąsiadujących punktów, można je zatem przedstawić w postaci:

$$G_R(j, k) = F(j, k) \otimes H_R(j, k)$$
(6)

$$G_C(j, k) = F(j, k) \otimes H_C(j, k)$$
(7)

gdzie  $H_R(j, k)$  i  $H_C(j, k)$  są maskami splotu (3x3) odpowiedzi (odpowiednio: w kolumnie i wierszu). Poniżej przedstawiono zapis macierzowy (dla masek splotu) dla omówionych algorytmów (pierwsza kolumna dla funkcji wiersza, druga dla kolumny).

$$\text{różnica sąsiednich punktów} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{różnica rozdzielona zerem} \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\text{Prewitt} \quad \frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Sobel} \quad \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\text{Frei-Chen} \quad \frac{1}{2+\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \quad \frac{1}{2+\sqrt{2}} \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

#### 4.2. Przykłady innych masek $H_R$ i $H_C$

Przedstawione poprzednio maski nie wyczerpują oczywiście wszystkich możliwości. Przykładem może być maska  $7 \times 7$  o współczynnikach:

$$H_R = \frac{1}{21} \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix} \quad (8)$$

lub jej zmodyfikowana wersja, w której współczynniki maleją liniowo wraz z odległością od punktu centralnego do brzegów:

$$H_R = \frac{1}{34} \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 2 & 2 & 0 & -2 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 2 & 0 & -2 & -2 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix} \quad (9)$$



### 4.3. Bardziej rozbudowane metody

Przedstawiona powyżej metoda może zostać odpowiednio zmodyfikowana poprzez wprowadzenie zamiast dwóch (pionowego i poziomego) ośmiu kierunków (np. góra, dół, lewo, prawo, góra-prawo, góra-lewo, dół-lewo, dół-prawo) wraz z odpowiednią funkcją sumującą. W tym przypadku kierunki traktowane dotychczas jako jeden (np. poziomy) zostały zastąpione przez dwa (lewo i prawo) ze względu na niesymetryczność maski splotu (por. maski przedstawione poniżej). Jeżeli kierunków jest dość dużo, można wybrać jako funkcję np. maksimum:

$$G(j, k) = \text{MAX} [|G_1(j, k)|, |G_2(j, k)|, \dots, |G_M(j, k)|] \quad (10)$$

gdzie:

$$G_m(j, k) = F(j, k) \otimes H_m(j, k) \quad \text{dla } m=1..M \quad (11)$$

Przykładowe maski splotu dla ośmiokierunkowego algorytmu są następujące (kolumny zawierają współczynniki wg: Prewitta, Kirscha, Robinsona):

prawo	$\frac{1}{5} \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\frac{1}{15} \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$
góra-prawo	$\frac{1}{5} \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$
góra	$\frac{1}{5} \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
góra-lewo	$\frac{1}{5} \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$
lewo	$\frac{1}{5} \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\frac{1}{15} \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$
dół-lewo	$\frac{1}{5} \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\frac{1}{15} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$

$$\begin{array}{l}
 \text{dół} \\
 \text{dół-prawo}
 \end{array}
 \begin{array}{l}
 \frac{1}{5} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} \\
 \frac{1}{5} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 \frac{1}{15} \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
 \frac{1}{15} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \\
 \frac{1}{3} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}
 \end{array}$$

Podobnie jak w przypadku dwukierunkowych algorytmów, również i w tym przypadku można zastosować maski o większych rozmiarach (np.  $5 \times 5$  lub  $7 \times 7$ ), wiąże się to jednak ze zwiększeniem złożoności obliczeniowej (a tym samym i czasu obliczeń).

Przedstawione powyżej metody stosują heurystyczne wartości współczynników w maskach splotu. Bardziej zaawansowane rozwiązania oparte są na wynikach analitycznego podejścia do problemu znajdowania krawędzi. Dobrym przykładem może być praca Canny'ego ([6]), w której przyjęto model krawędzi z rys. 2a z dodatkowym sygnałem szumu białego o standardowej dewiacji  $\sigma_n$ . Przyjęto tam założenie, że ciągły sygnał  $f(x)$  będzie splotem nieparzystej funkcji odpowiedzi  $h(x)$  w obszarze  $[-W, W]$ , zaś krawędź będzie lokalnym maksimum  $f(x) \otimes h(x)$  (dla którego  $f(x) \otimes h(x)$  jest równe zero). Aby wyznaczyć  $h(x)$ , zdefiniowane zostały następujące wskaźniki:

- poprawność wykrycia krawędzi – minimalizacja błędnego wskazania nieistniejącej krawędzi lub pominięcia krawędzi istniejącej za pomocą współczynnika sygnału do szumu (*signal-to-noise ratio*):

$$snr = \frac{h_E}{\sigma_n} S(h), \text{ gdzie} \quad (12)$$

$$S(h) = \frac{\int_{-W}^0 h(x) dx}{\int_{-W}^W [h(x)]^2 dx} \quad (13)$$

- poprawność lokalizacji krawędzi – wartość maksymalna po zastosowaniu operatora powinna być jak najbliższa środkowi przejścia pomiędzy wartością  $a$  i  $b$ . Wskaźnik lokalizacji został zdefiniowany jako:

$$LOC = \frac{h_E}{\sigma_n} L(h), \text{ gdzie} \quad (14)$$

$$L(h) = \frac{h'(0)}{\int_{-W}^W [h(x)]^2 dx}, \text{ gdzie } h'(x) \text{ jest pochodną } h(x) \quad (15)$$

- pojedyncza odpowiedź – w obszarze występowania krawędzi powinna zostać wykryta krawędź tylko jednokrotnie. Odległość pomiędzy dwoma maksimami, w przypadku



gdy występuje tylko sam szum, została zdefiniowana jako iloczyn szerokości obszaru  $W$  oraz współczynnika  $k$  ( $0 < k < 1$ ):

$$x_m = kW \tag{16}$$

W zależności od tych trzech współczynników otrzymujemy różne charakterystyki odpowiedzi.

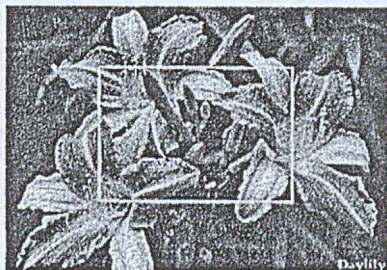
Ponieważ w algorytmie założono ciągłość funkcji, aby dostosować go do rzeczywistych warunków, należy przyjąć dość duże wartości  $W$  (duże okno), by zminimalizować obciążenie wyższych częstotliwości w skwantowanym sygnale wejściowym.

Rozwinięcie powyższej metody można znaleźć w [5], [6] oraz [8].

Oczywiście przedstawione powyżej algorytmy nie obejmują wszystkich rozwiązań mających na celu ekstrakcję krawędzi z obrazów, czego przykładem może być chociażby praca [7], w której wykorzystano elementy logiki rozmytej do ekstrakcji zmian jasności punktów oraz śledzenia przebiegu krawędzi w obrazie.

#### 4.4. Przykładowe wyniki działania algorytmów

Poniżej zostały przedstawione wyniki uzyskane po zastosowaniu algorytmów omówionych w poprzednich podrozdziałach na przykładzie obrazu z rys. 9 (dla obszaru zaznaczonego prostokątem).



Rys. 9. Obraz przed przekształceniami  
 Fig. 9. Original image

Na rys. 10. widzimy efekt działania dwóch najprostszych algorytmów wykrywania krawędzi: różnicy wartości dwóch sąsiednich punktów (z lewej) oraz różnicy rozdzielonej zerem (z prawej). Dla lepszego zobrazowania rezultatów została zmieniona paleta barw tak, by kolor czarny odpowiadał maksymalnej różnicy wartości, natomiast kolor biały oznaczał obszary jednolite.

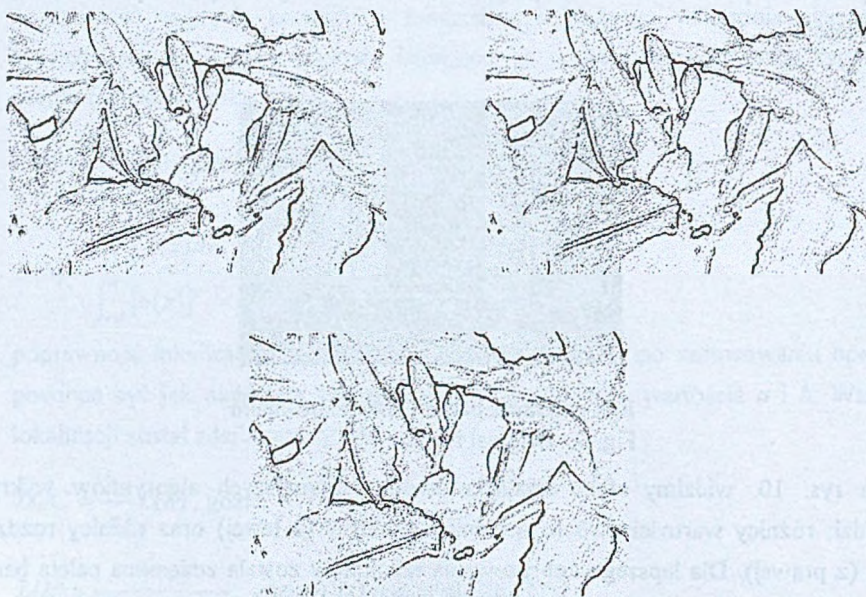




Rys. 10. Wyniki działania algorytmów (różnica punktów oraz różnica rozdzielona zerem)  
 Fig. 10. Results of image processing (pixel difference and separated pixel difference)

Z analizy obu obrazów widać, że lepsze właściwości wykrywania (wyraźniej zaznaczone krawędzie) ma pierwszy algorytm, jednak zgodnie z uwagą z rozdziału 4.1 powoduje to przesunięcie maksimum sygnału wyjściowego względem oryginalnego położenia krawędzi.

Zastosowanie bardziej skomplikowanych metod (podanych przez Prewitta, Sobela oraz Freia i Chena (przedstawione odpowiednio w lewej, prawej oraz w dolnej części rys. 11.) pozwalają na stwierdzenie, że chociaż najbardziej pożądane jest, aby we wszystkich kierunkach krawędzie były wykrywane z jednakową intensywnością, to w praktycznych zastosowaniach różnice między nimi są pomijalnie małe.



Rys. 11. Wyniki działania algorytmów (Prewitt, Sobel, Frei i Chen)  
 Fig. 11. Results of image processing (Prewitt, Sobel, Frei i Chen)

Zastosowanie algorytmów działających w większym otoczeniu punktu (podane w rozdziale 4.2) pozwala na jeszcze lepszą (niż w przypadku poprzednich algorytmów)



detekcję krawędzi (rys. 12 - maska (8) - obraz z lewej, maska (9) - obraz z prawej), a przy tym jest zdecydowanie mniej podatne na występujące w obrazie szumy.

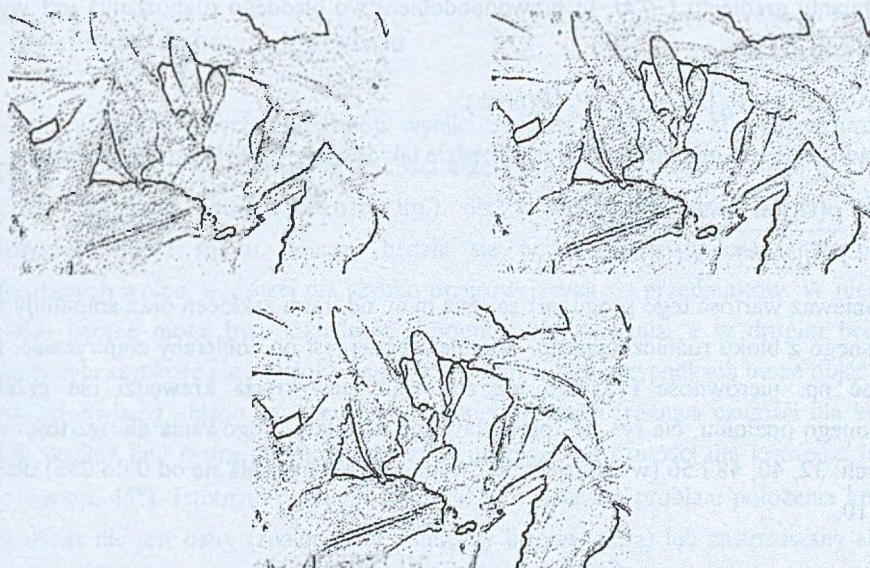
W przypadku zastosowania algorytmów działających w więcej niż w dwóch



Rys. 12. Wyniki działania algorytmów (maski 7x7)

Fig. 12. Results of image processing (7x7 masks)

(ortogonalnych) kierunkach (podanych w rozdziale 4.3 - rys. 13) otrzymujemy wyniki nieco lepsze niż w przypadku algorytmów najprostszych, ale zdecydowanie gorsze niż w przypadku maski 7x7 (por. rys. 10, 11, 12). W przypadku tych algorytmów (8 kierunków, maski 3x3) złożoność obliczeniowa jest niewiele mniejsza niż dla algorytmów z maskami ortogonalnymi 7x7.



Rys. 13. Wyniki działania algorytmów (maski ośmiokierunkowe)

Fig. 13. Results of image processing (eight-directional masks)

Z przedstawionego porównania wynika, że wybór metody zależy głównie od wymaganej dokładności detekcji krawędzi, jak również od takich czynników, jak szybkość działania (czas przeliczenia obrazu) oraz odporność na szumy.



## 5. Progowanie

Zgodnie ze schematem przedstawionym na rys. 6 po wyznaczeniu zmian w jasności sąsiadujących punktów należy dokonać selekcji pomiędzy tłem obrazu a szukanymi krawędziami. W przypadku obrazu wolnego od szumów próg, którego przekroczenie jest uznawane za wystąpienie krawędzi, można dość łatwo wyznaczyć (choćby na podstawie średniej jasności całego obrazu). Sprawa komplikuje się, gdy w obrazie pojawią się szumy i dla takiego obrazu należy tak dobrać poziom detekcji, by prawdopodobieństwo błędu (przeoczenie istniejącej krawędzi lub znalezienie nieistniejącej) było minimalne.

Jeżeli  $P(jest)$  oznacza prawdopodobieństwo wystąpienia krawędzi, a  $P(brak)$  prawdopodobieństwo niewystąpienia ([9]), niech

$$P_D = \int_1^{\infty} p(G|jest) dG \quad (17)$$

$$P_F = \int_1^{\infty} p(G|brak) dG \quad (18)$$

oznaczają odpowiednio poprawną i niepoprawną detekcję, zaś  $p(G|jest)$  i  $p(G|brak)$  dystrybuanty gradientu  $G(j,k)$ , to prawdopodobieństwo błędnego rozpoznania jest wyrażone równaniem:

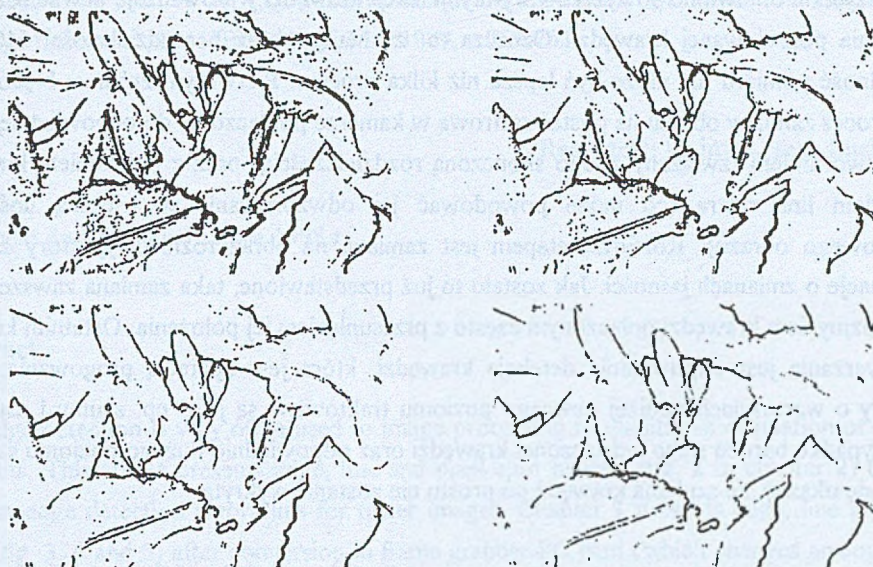
$$P_E = [1 - P_D]P(jest) + [P_F]P(brak) \quad (19)$$

Błąd ten będzie minimalny, jeśli próg będzie tak dobrany, by był spełniony warunek:

$$\frac{p(G|jest)}{p(G|brak)} \geq \frac{P(G|brak)}{P(G|jest)} \quad (20)$$

Ponieważ wartość tego progu jest zależna m.in. od ilości zakłóceń oraz amplitudy sygnału uzyskanego z bloku różniczkującego, więc najczęściej jest on dobierany empirycznie, tak aby uzyskać np. nierówność (15) lub też aby błąd niewykrycia krawędzi nie przekroczył określonego poziomu. Na rys. 14. przedstawiono rezultaty progowania dla wartości progów równych: 32, 40, 48 i 56 (w obrazie źródłowym jasność zmieniała się od 0 do 255) dla obrazu z rys. 10.





Rys. 14. Wyniki działania algorytmu progowania dla progu równego: 32, 40, 48 i 56  
 Fig. 14. Results of threshold selection for threshold: 32, 40, 48 and 56

## 6. Problemy doboru algorytmu

Wielość przedstawionych algorytmów wynika z różnych wymagań stawianych procesowi detekcji. W zależności od wymaganej jakości algorytmu (prawdopodobieństwa poprawnego znalezienia krawędzi, jego szybkości itp.) dla każdego problemu dobiera się inne współczynniki maski splotu. Inaczej będzie się przedstawiała realizacja dla obrazów przychodzących wolno, a inaczej dla szybko przesuwających się przedmiotów. W pierwszym przypadku istotne mogą być dokładność i poprawność działania, a w drugim będzie się bardziej liczyła szybkość nie zaś dokładność. Poza tym w obu przypadkach może dojść jeszcze problem odpowiedzi układu na orientację szukanej krawędzi (różnica czułości dla krawędzi leżących wzdłuż linii rastra nie może znacząco odbiegać od czułości dla krawędzi leżących pod kątem np.  $45^\circ$ ). Istotnym punktem może także okazać się problem położenia krawędzi, gdy jej obraz nie jest ostry (znajduje się pomiędzy liniami rastra) lub zastosowany algorytm powoduje rozmycie lub przesunięcie jej obrazu.

Kolejnym ważnym problemem jest oczywiście wpływ szumów na ostateczny rezultat działania algorytmu. W zależności od ich wartości należy tak dobrać rodzaj algorytmu, by skuteczność (prawdopodobieństwo poprawnej detekcji) była na zadowalającym poziomie.



Wszystkie omówione powyżej algorytmy detekcji krawędzi wprowadzają pewne rozmycie położenia poszukiwanej krawędzi. Oznacza to, że mając obraz o rozdzielczości 640x480, dokładność pomiaru nie może być lepsza niż kilka procent. Pierwszym źródłem błędów jest sam proces zamiany obrazu na postać rastrową w kamerze podłączonej do odpowiedniej karty z przetwornikiem (związany z jego skończoną rozdzielczością) oraz z położeniem krawędzi względem linii rastra (co może powodować jej odwzorowanie w większą ilość linii wynikowego obrazu). Kolejnym etapem jest zamiana na obraz różnicowy, który zawiera informacje o zmianach jasności. Jak zostało to już przedstawione, taka zamiana zawsze wiąże się z rozmyciem krawędzi połączonym często z przesunięciem jej położenia. Ostatnim krokiem przetwarzania jest odpowiednia detekcja krawędzi, która jest operacją progowania, gdzie sygnały o wartościach poniżej pewnego poziomu traktowane są jako np. szумы i usuwane. W przypadku bardzo słabo zaznaczonej krawędzi oraz odpowiednio dużego poziomu szumów może się okazać, że szukana krawędź po prostu nie zostanie wykryta.

## LITERATURA

1. Prewitt J. M. S.: „Object Enhancement and Extraction”, *Picture Processing and Psychopictorics*, B.S. Lipkin and A Rosenfeld, Eds. Academic Press, New York 1970.
2. Duda R. O., Hart P. E.: „Pattern Classification and Scene Analysis”, Wiley, New York 1973.
3. Frei I., Chen C.: „Fast Boundary Detection: A Generalization and a New Algorithm”, *IEEE Transactions Computers*, C-26, 10, October 1977, s. 988-998.
4. Canny J.: „A Computational Approach to Edge Detection”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-8, 6 Nov 1986.
5. Tagare H. D., deFigueiredo R. J. P.: „On the Localization Performance Measure and Optimal Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, December 1990, s. 1186-1190.
6. Sarkar S., Boyer K. L.: „On Optimal Infinite Impulse Response Edge Detection Filters”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, November 1991, s. 1154-1171.
7. Law T., Itoh H., Seki H.: „Image Filtering, Edge Detection, and Edge Tracing Using Fuzzy Reasoning”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, May 1996, s. 481-491.
8. Demign D., Kamle T.: „A Discrete Expression of Canny's Criteria for Step Edge Detector Performances Evaluation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, November 1997, s. 1199-1211.



9. Abdou I. E., Pratt W. K.: „Quantitative Design and Evaluation of Enhancement.Thresholding Edge Detectors”, Proc IEEE, 67, 5, May 1979, s. 753-763.

Recenzent: Dr inż. Leszek Luchowski

Wpłynęło do Redakcji 13 maja 1998 r.

### Abstract

Edge detection is very often used to image processing to qualitative estimation of picture contents. This article presents edge, line and pixel spot models (fig. 2 in chapter 2) used to analyse edge detection algorithms for raster images. Chapter 3 presents edge, line and pixel spot (fig. 3, 4 and 5) after conversion in frame grabber PC card (which changes analog signal from CCD camera to digital discrete matrix of numbers).

The edge detection algorithm consists of two steps (fig. 6). The first step is spatial differentiation which changes original image to image that contains gradient amplitude. In the second step each image pixel is compared to a threshold value.

First part of this article presents simple convolution masks known from literature (e.g. presented by Prewitt, Sobel, Frei and Chen) and more difficulty edge detection presented by Canny. Main part of this paper compares algorithms presented in chapter 4. Figure 10, 11, 12 and 13 show effects after use each of method according to original images presented on fig. 9.

The chapter 5 is a short discussion about optimal selection of threshold value. Images on fig. 14 show results for threshold value 32, 40, 48 and 56 (for pixel value range from 0 to 255).