

Andrzej ORDYS  
Konrad WOJCIECHOWSKI

WYBRANE ALGORYTMY PRZETWARZANIA WSTĘPNEGO  
OBRAZU W ZAGADNIENIU WYZNACZANIA POLA PRĘDKOŚCI<sup>x)</sup>

**Streszczenie.** W pracy przedstawiono algorytmy przetwarzania wstępnego obrazu, które są użyteczne przy określaniu pola prędkości. Przyjmuje się, że pole prędkości będzie obliczane metodą minimalizacji wariancji wektora prędkości wzdłuż konturu obiektu. Prezentowane algorytmy związane są dwoma różnymi metodami liczenia składowej prostopadłej wektora prędkości. Metoda pierwsza bazuje na obliczaniu przemieszczenia konturu. Metoda druga wykorzystuje zmianę pewnej funkcji poziomu szarości na konturze.

## 1. Wstęp

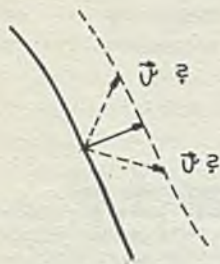
Pole prędkości jest użytecznym narzędziem pozwalającym na określanie parametrów ruchu względnego obserwatora i obrazu. Jest ono wykorzystywane głównie jako model mechanizmu detekcji ruchu przez organizmy żywe (prace [4], [5], [12]). Sugeruje to możliwość wykorzystania pola prędkości także w układach sztucznej inteligencji, które wypracowują decyzję sterującą na podstawie odbieranej informacji wizyjnej.

Przez pole prędkości rozumiemy przyporządkowanie każdemu punktowi obrazu wektora prędkości w tym punkcie. Wiele prac (np. [1], [3], [10]) poświęconych jest badaniu związków między wyglądem pola prędkości i parametrami ruchu względnego obserwatora i obrazu. Rozważa się różne kształty obiektów na obrazach. W większości przypadków konieczna jest znajomość a priori wektora prędkości w pewnych punktach obrazu dla odtworzenia całego pola prędkości.

W ostatnim czasie rozwijana jest technika obliczania pola prędkości, która nie wymaga znajomości a priori wektorów prędkości. Metoda ta nazywana jest metodą "wygładzania pola prędkości" lub "minimalnej wariancji pola prędkości" [7], [8]. Jest to metoda przybliżona jednakże szereg przykładów [7] potwierdza jej przydatność w przypadku niewielkiego odstępu czasu  $\Delta t$  między kolejnymi obserwowanymi obrazami. Idea metody opiera się na sposobie, że obserwacja przesuwających się obrazów dostarcza informacji

x) Praca finansowana z Centralnego Programu Badań Podstawowych CPBP 02.13 "Układy ze sztuczną inteligencją do maszyn roboczych i pojazdów".

o składowej wektora prędkości prostopadłej do konturu. Składowa prędkości równoległa do konturu w wielu przypadkach jest nieokreślona. Ilustruje to rys. 1. Na rysunku tym linia przesuwa się do położenia zaznaczonego kropkami. Obraz nie dostarcza informacji o składowej wektora prędkości równoległej do konturu.



Rys. 1. Przykład sytuacji, w której składowa wektora prędkości równoległa do konturu nie jest określona

Fig. 1. The component parallel to the contour is not determined

W metodzie minimalnej wariacji zakłada się, że składowa prostopadła wektora prędkości jest znana. Składową równoległą dobiera się w ten sposób, by wariacja wektora prędkości liczona wzdłuż konturu była minimalna. Oznacza to rozwiązanie następującego problemu optymalizacji statycznej: zminimalizować:

$$I = \int_S \left| \frac{\partial \vec{v}}{\partial s} \right|^2 ds \quad (1)$$

przy ograniczeniu:

$$(\vec{v} \cdot \vec{n}) = v^n, \quad (2)$$

gdzie:

$\vec{v}$  - wektor prędkości,

$\vec{n}$  - wektor jednostkowy prostopadły do konturu,

$v^n$  - długość składowej prostopadłej prędkości,

$S$  - kontur.

Rozwiązanie problemu można znaleźć w pracy [13].

Wymaganie znajomości składowej prostopadłej wektora prędkości jest słabsze niż żądanie znajomości całego wektora prędkości w pewnych punktach (dowodzi tego rys. 1). Jednakże składowa prostopadła także nie jest znana dokładnie.

Przedstawione w pracy algorytmy służą do wyznaczenia składowej prostopadłej wektora prędkości w sposób przybliżony. Pierwszy z nich oblicza składową prostopadłą wektora prędkości jako iloraz odległości między obrazami przez odstęp czasu  $\Delta t$ . Wykonuje się więc następujące operacje: konturowanie obrazu, liczenie kierunku prostopadłego do konturu w każdym

jego punkcie, liczenie przesunięcia obrazu w tym kierunku. Drugi algorytm wykorzystuje do detekcji  $v^n$  zmianę stopnia szarości na konturze w czasie. Obraz poddawany jest wstępnemu przetwarzaniu, po czym liczona jest pochodna funkcji obrazu po czasie. Algorytm wymaga także znajomości gradientu funkcji obrazu.

Przedstawiane algorytmy mają zastosowanie tylko wówczas, gdy przemieszczenie obrazu jest niewielkie, co oznacza małe odstępy czasu  $\Delta t$  między obserwowanymi obrazami. Zastosowanie ich do przypadku znacznego przesunięcia obrazów może prowadzić do zupełnie błędnych wyników.

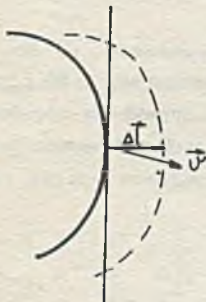
## 2. Obliczanie pola prędkości na podstawie przesunięcia konturu

Rozważmy fragment obrazu przedstawiony na rys. 2. Położenie pierwotne zaznaczone jest linią ciągłą, natomiast położenie po przemieszczeniu linią kropkowaną. Z rysunku widać, że dla małych przemieszczeń składowa prostopadła wektora prędkości może być wyznaczona przez wydzielenie przemieszczenia w kierunku prostopadłym przez  $\Delta t$ .

Wymaga to wykonania następujących operacji:

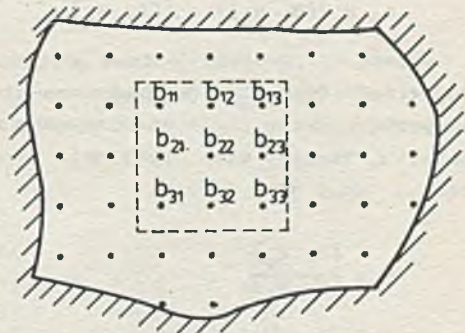
- konturowanie obrazu,
- obliczanie kierunku prostopadłego do konturu w każdym jego punkcie,
- obliczanie przesunięcia konturu w tym kierunku.

Algorytm konturowania powinien być tak dobrany, by pozwalał w łatwy sposób poliożyć następnie kierunek prostopadły do konturu. Warunek ten spełniają dwa przedstawione poniżej algorytmy [6].



Rys. 2. Dla małych przemieszczeń  $\frac{\Delta I}{\Delta t}$  dobrze aproksymuje składową prostopadłą wektora prędkości

Fig. 2. Method of determining the component orthogonal to the contour. For small displacements  $\frac{\Delta I}{\Delta t}$  gives a good approximation.



Rys. 3. Ramka o wymiarze 3 x 3 na powierzchni obrazu  $b_{ij}$  oznacza poziom szarości w punkcie  $(ij)$

Fig. 3. Mask 3 x 3 pixel on the image area  $(b_{ij})$  means grey level at the point  $(i,j)$

### Algorytm uśrednionego gradientu (AG)

Przyjmujemy, że po obrazie przesuwana się ramka o wymiarach  $m \times n$ . W każdym swoim położeniu przykrywa ona więc  $m \cdot n$  punktów obrazu. Zostało to pokazane na rys. 3. Oznaczmy przez  $B$  macierz zawierającą wartości poziomów szarości dla punktów znajdujących się wewnątrz ramki:

$$B = \begin{bmatrix} b_{11} & \dots & b_{1m} \\ \vdots & & \vdots \\ b_{n1} & \dots & b_{nm} \end{bmatrix} \quad (3)$$

Wprowadzmy oznaczenie:

$$(B, C) = \sum_{i=1}^n \sum_{j=1}^m b_{ij} \cdot c_{ij} \quad (4)$$

Dla każdego położenia ramki na obrazie należy obliczyć wyrażenie:

$$z_1 = \left[ (B, W_1)^2 + (B, W_2)^2 \right]^{1/2} \quad (5)$$

lub też prostsze wyrażenie:

$$z_2 = |(B, W_1)| + |(B, W_2)| \quad (6)$$

Jeśli  $z_1$  (lub  $z_2$ ) jest większe niż arbitralnie wybrana wartość progowa, to przyjmuje się, że w analizowanym fragmencie obrazu znajduje się brzeg obiektu. Tangens kąta nachylenia stycznej do konturu w tym fragmencie obrazu dany jest wzorem:

$$k = \frac{(B, W_2)}{(B, W_1)} \quad (7)$$

$W_1, W_2$  są funkcjami wagowymi. Przykłady takich funkcji podano niżej:

a) gradient Roberta

$$W_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad W_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (8)$$

b) wygładzony gradient

$$W_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad W_2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (9)$$

o) funkcje wagowe Sobela

$$W_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad W_2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (10)$$

d) izotropowe funkcje wagowe

$$W_1 = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad W_2 = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \quad (11)$$

Algorytm używający zorientowanych szablonów

Ustala się zbiór szablonów  $\{T_j\}$ . Dla każdego z położeń ramki oblicza się:

$$q = \max_1 \{(B, T_1)\} \quad (12)$$

Gdy wartość  $q$  przekroczy arbitralnie przyjętą wartość progową, to przyjmuje się, że w analizowanym fragmencie obrazu znajduje się brzeg obiektu. Jego orientacja określona jest przez orientację szablonu  $T_j$ , dla którego wystąpiło maksimum.

Przykład zbioru szablonów:

$$\begin{aligned} T_1 &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}; & T_2 &= \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}; & T_3 &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 2 & 1 & 0 \end{bmatrix}; \\ T_4 &= \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}; & T_5 &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; & T_6 &= \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}; \\ T_7 &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; & T_8 &= \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \end{aligned} \quad (13)$$

Jak widać, każdy z szablonów określa inny kierunek konturu obiektu. Kierunki te są wielokrotnością kąta  $\frac{\pi}{4}$ . Stąd dokładność określenia kierunku wynosi  $\pm \frac{\pi}{8}$ .

#### Algorytm wykrywający zmianę poziomu szarości

Algorytm ten nie określa kierunku konturu. Zostanie przedstawiony dla przeciwieństwa dwóm wcześniejszym. Nadaje się on do obrazów o ostro zarysowych krawędziach obiektów, najlepiej do obrazów o dwóch poziomach szarości. Algorytm polega na przebieganiu kolejno (wierszami) wszystkich punktów obrazu i zapamiętywaniu tych punktów, w których nastąpiła zmiana poziomu szarości o więcej niż zadana wartość progowa

$$b_i - b_{i-1} > \Delta \quad (14)$$

Punkty te tworzą kontur obiektu.

Należy zauważyć, że nie jest wykonywana filtracja obrazu. Przypadkowe błędy w wartości poziomu szarości zostaną zidentyfikowane jako punkty konturu i w związku z tym nawet pogrubione..

#### Obliczanie kierunku prostopadłego do konturu w danym punkcie

Algorytmy konturowania opisane wyżej (z wyjątkiem ostatniego) obliczają dodatkowo kąt nachylenia stycznej do konturu w każdym jego punkcie. Kierunek normalny do konturu jest więc natychmiast znany. Dla algorytmu uśrednionego gradientu tangens kąta nachylenia normalnej do konturu wynosi:

$$n = \frac{(B, W_1)}{(B, W_2)} \quad (15)$$

W algorytmie używającym zorientowanych szablonów określa się nie tylko kierunek wektora normalnego do konturu, ale także jego zwrot na zewnątrz konturu. Mała dokładność określenia kierunku stanowi wadę tej metody. W celu zwiększenia dokładności należałoby powiększyć wymiary ramki.

Dla zestawu szablonów takiego jak podany wyżej kąt, jaki tworzy normalna do konturu z osią  $Ox$  układu współrzędnych, dany jest wzorem:

$$\alpha = -\frac{3}{4} + j \frac{\pi}{4} \quad (16)$$

$j$  oznacza numer szablonu, dla którego wystąpiło maksimum wyrażenia  $(B, T_i)$  dla  $i = 1, \dots, 8$ .

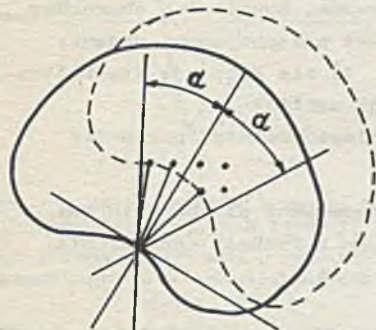
Obliczenie wielkości przesunięcia w kierunku prostopadłym do konturu

Zakładamy, że dysponujemy dwoma obrazami i odpowiadającymi im dwoma konturami - pierwotnym i przesuniętym. Oznaczmy przez  $(x_i, y_i)$  punkt konturu pierwotnego a przez  $(x'_i, y'_i)$  punkt konturu przesuniętego. Niech będzie dany punkt konturu pierwotnego  $(x_j, y_j)$  oraz kąt nachylenia normalnej do konturu w tym punkcie  $\alpha_j$ . Spośród punktów  $(x'_i, y'_i)$  konturu przesuniętego (gdzie  $i = 1, \dots, K$  a  $K$  oznacza liczbę punktów konturu) wybieramy te, dla których zachodzi:

$$\left| \frac{y'_i - y_j}{x'_i - x_j} - \operatorname{tg} \alpha_j \right| < p \quad (17)$$

$p$  - arbitralnie wybrany współczynnik wagowy. Z punktów spełniających warunek na kąt wybieramy ten, którego odległość od punktu  $(x_j, y_j)$  jest najmniejsza. Odległość wyraża się wzorem:

$$d = \left[ (y'_i - y_j)^2 + (x'_i - x_j)^2 \right]^{1/2} \quad (18)$$



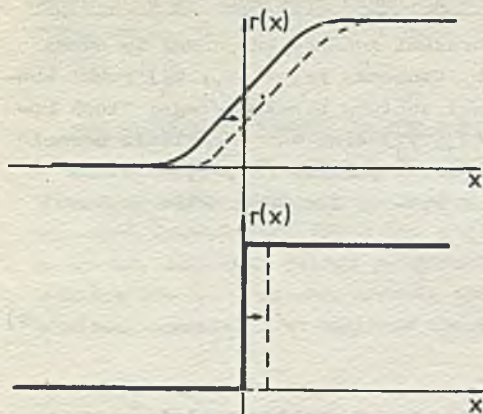
Rys. 4. Obliczanie wartości przesunięcia w kierunku prostopadłym do konturu. Kropkami zaznaczono punkty obrazu

Fig. 4. Determination of the displacement on the direction orthogonal to the contour

Minimalna wartość  $d$  jest szukanym przesunięciem w kierunku prostopadłym. Ideę algorytmu ilustruje rysunek 4. Jak widać z rysunku 4 wyznaczona wartość  $d$  dobrze aproksymuje rzeczywistą wartość przesunięcia w kierunku prostopadłym. Dobór kąta  $\alpha$  stanowi kompromis między żądaniem dokładności określenia  $d$  oraz koniecznością trafienia w punkt na przesuniętym konturze. Algorytm zakłada niewielkie przesunięcie obrazu takie, by między kontur i kontur przesunięty nie weszła żadna inna linia.

3. Obliczanie pola prędkości na podstawie funkcji zmian szarości

Metoda ta korzysta z funkcji szarości określonej na obrazie. Dokonuje się transformacji tej funkcji używając operatora, który nazywany jest laplasjanem z funkcji Gaussa. Następnie oblicza się zmianę przekształconej funkcji szarości w czasie dla punktów konturu obiektu. Na tej podstawie wnioskuje się o składowej prostopadłej wektora prędkości tych punktów. Metoda ta wymaga większego nakładu obliczeń niż algorytm przedstawiony w rozdziale 2. Dla bardziej poglądowego wyjaśnienia problemu zostanie najpierw przedstawiony przypadek obrazu jednowymiarowego, następnie wyniki zostaną uogólnione na przypadek dwuwymiarowy.



Rys. 5. Zmiana stopnia szarości na brzegu przedmiotu:

a)  $r(x)$  jest ciągłą funkcją  $x$ , b)  $r(x)$  jest dwuwartościową funkcją  $x$

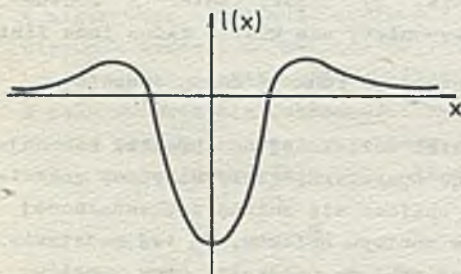
Fig. 5. Gray-level function on the edge - one dimension case

2) duża stromość funkcji ogranicza stosowalność metody (przypadek skrajny daje rysunek 5b).

W pracy [9] zaproponowano jako operator odpowiedni dla badania zmian stopnia szarości na brzegu przedmiotu laplasjan z funkcji Gaussa. Dla przypadku jednowymiarowego funkcja Gaussa ma postać:

$$g(x) = b \cdot \exp(-ax^2) \quad (19)$$

Laplasjan z funkcji  $g(x)$  dany jest wzorem:



Rys. 6. Laplasjan z funkcji Gaussa

Fig. 6. Laplasjan of gaussian - one dimension case

Rysunek 5 przedstawia przykładowe funkcje poziomu szarości  $r(x)$  na brzegu przedmiotu na obrazie jednowymiarowym. Rysunek 5a odpowiada ciągłym zmianom stopnia szarości, natomiast na rysunku 5b mamy do czynienia z obrazem o dwóch poziomach szarości. W przypadku takim jak na rys. 5a zmiana stopnia szarości w punkcie zero może być określona bezpośrednio z rysunku - przez porównanie wartości funkcji zaznaczonej linią ciągłą z wartością zaznaczoną linią przerywaną. Sposób ten obaroczony jest następującymi wadami:

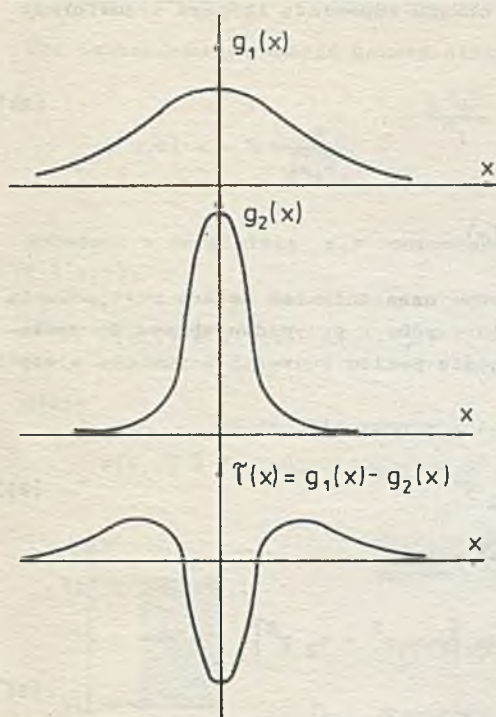
1) nie dokonuje się filtracji zakłóceń,

$$\begin{aligned} l(x) &= \frac{d^2}{dx^2} g(x) = \\ &= 2ba \cdot \exp(-ax^2) \cdot \\ &\cdot (2ax^2 - 1) \end{aligned} \quad (20)$$

Kształt funkcji  $l(x)$  przedstawia rys. 6.

Funkcja  $l(x)$  może być aproksymowana przez różnicę dwóch funkcji Gaussa, tak jak to pokazano na rys. 7.

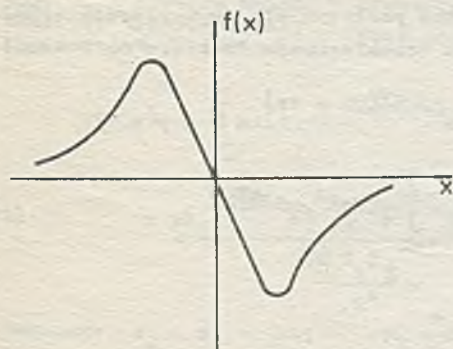




Rys. 7. Aproksymacja funkcji  $l(x)$  przez różnicę dwóch funkcji Gaussa:

$a, b$  - funkcja Gaussa o różnych współczynnikach,  $c$  - różnica funkcji z rysunków 7a i 7b

Fig. 7. Approximation of the laplasjen of gaussian by the difference of two Gauss' functions



Rys. 8. Splot funkcji szarości  $r(x)$  z rys.5 z drugą pochodną funkcji Gaussa

Fig. 8. Convolution of grey-level function from the picture 5b with second derivative of Gauss' function

Przekształconą funkcję szarości uzyskuje się przez wykonanie splotu funkcji szarości  $r(x)$  z laplasjanem Gaussa  $l(x)$ .

$$f(x) = \int_{-\infty}^{\infty} r(t) \cdot l(x - t) dt \quad (21)$$

Funkcja  $f(x)$  na ogół nie może być policzona analitycznie. Jej wygląd, dla funkcji  $r(x)$  takich jak na rys. 5, pokazany jest na rys. 8.

Użytecznym aparatem matematycznym przy określaniu funkcji  $f(x)$  jest transformacja Fouriera. Transformacja Fouriera funkcji  $l(x)$  może być wyznaczona analitycznie, natomiast dla funkcji  $r(x)$  wykorzystać można metodę szybkiej transformacji Fouriera (FFT).

W celu określenia  $L(j\omega)$  dokonamy najpierw transformacji funkcji Gaussa:

$$\begin{aligned} G(j\omega) &= \int_{-\infty}^{\infty} e^{-j\omega x} g(x) dx = \\ &= \int_{-\infty}^{\infty} e^{-j\omega x} b e^{-ax^2} dx = \\ &= b \sqrt{\frac{\pi}{a}} e^{-\left(\frac{\omega^2}{4a}\right)} \quad (22) \end{aligned}$$

Ponieważ  $l(x)$  jest drugą pochodną funkcji  $g(x)$ , więc  $L(j\omega)$  dane jest wzorem:

$$\begin{aligned} L(j\omega) &= -\omega^2 G(j\omega) = \\ &= -\omega^2 b \sqrt{\frac{\pi}{a}} \cdot \\ &\cdot \exp\left(-\frac{\omega^2}{4a}\right) \quad (23) \end{aligned}$$

Splotowi w dziedzinie współrzędnej obrazu odpowiada iloczyn transformat Fouriera funkcji  $l(x)$  i  $r(x)$ .

$$F(j\omega) = L(j\omega) R(j\omega), \quad (24)$$

gdzie

$$F(j\omega) \hat{=} f(x) \quad R(j\omega) \hat{=} r(x).$$

Przedstawione rozważania są poglądowym uzasadnieniem metody postępowania jaką stosuje się do obrazów dwuwymiarowych. W przypadku obrazu dwuwymiarowego funkcja  $r(x,y)$  określać będzie poziom szarości w punkcie o współrzędnych  $(x,y)$ .

Przyjmijemy dwuwymiarową funkcję Gaussa o postaci:

$$g(x,y) = b \cdot \exp \left[ -a_1 x^2 - a_2 y^2 \right] \quad (25)$$

Laplasjan funkcji  $g(x,y)$  dany jest wzorem:

$$l(x,y) = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} = 2 b \cdot \exp \left[ -a_1 x^2 - a_2 y^2 \right] \cdot \left\{ a_1 (2a_1 x^2 - 1) + a_2 (2a_2 y^2 - 1) \right\} \quad (26)$$

Podobnie jak dla obrazu jednowymiarowego przetworzenie obrazu polega na dokonaniu splotu funkcji  $r(x,y)$  i  $l(x,y)$ :

$$f(x,y) = \iint_{-\infty}^{\infty} r(t,\tau) l(x-t, y-\tau) dt d\tau \quad (27)$$

Zamiast obliczania splotu funkcji można posłużyć się dwuwymiarową transformacją Fouriera. Dla funkcji Gaussa transformacja Fouriera ma postać:

$$\begin{aligned} G(u,v) &= \iint_{-\infty}^{\infty} b e^{-a_1 x^2 - a_2 y^2} e^{-j2\pi(ux + vy)} dx dy = \\ &= b \int_{-\infty}^{\infty} e^{-a_1 x^2} e^{-j2\pi ux} dx \int_{-\infty}^{\infty} e^{-a_2 y^2} e^{-j2\pi vy} dy = \\ &= b \frac{\pi}{\sqrt{a_1 a_2}} e^{-\frac{\pi^2 u^2}{a_1}} \cdot e^{-\frac{\pi^2 v^2}{a_2}} \end{aligned} \quad (28)$$

Dla laplasjanu z funkcji Gaussa otrzymujemy:

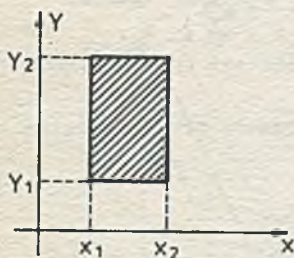
$$L(u,v) = -b \frac{\mathcal{I}}{\sqrt{a_1 a_2}} e^{-\frac{\mathcal{I}^2 u^2}{a_1}} e^{-\frac{\mathcal{I}^2 v^2}{a_2}} (u^2 + v^2) \quad (29)$$

Splotowi w dziedzinie  $x, y$  odpowiada iloczyn transformat funkcji  $r(x, y)$  i  $l(x, y)$ :

$$F(u, v) = R(u, v) L(u, v), \quad (30)$$

gdzie

$$F(u, v) \hat{=} f(x, y), \quad R(u, v) \hat{=} r(x, y).$$



Rys. 9. Przykład obrazu, dla którego można wyznaczyć analitycznie transformatę Fouriera

Fig. 9. A picture which Fourier transform may be computed analitically

Kształt funkcji  $f(x, y)$  otrzymamy dokonując odwrotnej transformacji Fouriera funkcji  $F(u, v)$ :

$$f(x, y) = \iint_{-\infty}^{\infty} F(u, v) e^{j2\mathcal{I}(ux+vy)} du dv \quad (31)$$

Transformacji funkcji  $r(x, y)$  można dokonać wykorzystując technikę szybkiej transformacji Fouriera (FFT). Dla bardzo prostych przypadków może się udać znalezienie transformaty analitycznie. Rozważmy przykładowo na białym tle czarny prostokąt o bokach zorientowanych równolegle do ramki obrazu (rysunek 9).

Transformacja Fouriera obrazu z rysunku 9 dana jest wzorem:

$$\begin{aligned} F(u, v) &= \iint_{-\infty}^{\infty} r(x, y) e^{-j2\mathcal{I}(ux+vy)} dx dy = \int_{x_1}^{x_2} e^{-j2\mathcal{I}ux} dx \int_{y_1}^{y_2} e^{-j2\mathcal{I}vy} dy = \\ &= \frac{e^{-j2\mathcal{I}ux_2} - e^{-j2\mathcal{I}ux_1}}{-j2\mathcal{I}u} \cdot \frac{e^{-j2\mathcal{I}vy_2} - e^{-j2\mathcal{I}vy_1}}{-j2\mathcal{I}v} \end{aligned} \quad (32)$$

Oznaczmy  $x_2 = x_1 + \Delta x$ ,  $y_2 = y_1 + \Delta y$ . Otrzymamy wówczas:

$$F(u, v) = e^{-j2\mathcal{I}ux_1} \cdot \frac{e^{-j\mathcal{I}u\Delta x} - e^{j\mathcal{I}u\Delta x}}{-j2\mathcal{I}u} \cdot \frac{e^{-j\mathcal{I}v\Delta y} - e^{j\mathcal{I}v\Delta y}}{-j2\mathcal{I}v}$$

$$\begin{aligned}
 & \cdot e^{-j2\pi v y_1} \frac{e^{-j\pi v \Delta y} \quad e^{j\pi v \Delta y} \quad e^{-j\pi v \Delta y}}{e^{-j2\pi v}} = \\
 & = e^{-j2\pi (u x_1 + v y_1)} \cdot \frac{\sin \pi \Delta x}{\pi u} \cdot e^{-j\pi u \Delta x} \cdot \frac{\sin \pi \Delta y}{\pi v} e^{-j\pi v \Delta y}
 \end{aligned} \quad (33)$$

Dotychczasowe rozważania prowadzone były dla obrazów o charakterze ciągłym, to znaczy wartości  $x, y$  przebiegały zbioru ciągłe. W rzeczywistości obrazy mają najczęściej charakter dyskretny. Wynika to z własności kamer stosowanych do odczytu obrazów oraz z faktu, że dane są pamiętane w maszynie cyfrowej. Rozważania dla przypadku dyskretnego nie różnią się jakościowo od tego co zostało przedstawione wyżej. W miejsce ciągłej funkcji szarości  $r(x, y)$  wprowadzimy dyskretną funkcję szarości  $r(m, n)$  oraz określimy dyskretną wersję laplasjanu z funkcji Gaussa  $l(m, n)$  jako wynik próbkowania  $l(x, y)$  w punktach  $x_m = \Delta x m$ ,  $y_n = \Delta y n$ , gdzie  $m = 0, \dots, M-1$ ,  $n = 0, \dots, N-1$ ,  $x = \frac{x}{M}$ ;  $y = \frac{y}{N}$ ;  $l_x, l_y$  oznaczają długości obrazu w osiach  $x$  i  $y$ ;  $M, N$  są liczbami punktów obrazu w osiach  $x$  i  $y$ .

Splot funkcji  $r(m, n)$  z laplasjanem z funkcji Gaussa przyjmie postać:

$$f(m, n) = \sum_{i=0}^{M-1} \sum_{k=0}^{N-1} r(i, k) l(m-i, n-k) \quad (34)$$

Podobnie jak dla przypadku ciągłego można obliczyć  $f(m, n)$ , wykorzystując transformację Fouriera. Dyskretna wersja transformacji Fouriera zdefiniowana jest przez następujące równanie:

$$F(u, v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} r(m, n) e^{-j2\pi \left( \frac{um}{M} + \frac{vn}{N} \right)} \quad (35)$$

$$u = 0, \dots, M-1, \quad v = 0, \dots, N-1$$

Splotowi w dziedzinie  $m, n$  odpowiada iloczyn transformatał:

$$F(u, v) = R(u, v) L(u, v). \quad (36)$$

Wreszcie transformacja odwrotna dana jest wzorem:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)} \quad (37)$$

Obliczanie składowej prostopadłej wektora prędkości dla punktów konturu

Dla ilustracji metody zajmijmy się przypadkiem jednowymiarowym. Rysunek 10a przedstawia funkcję  $f(x)$  (zaznaczoną linią ciągłą) oraz zaznaczone liniami przerywanymi, przesunięcia funkcji  $f(x)$  (w lewo i w prawo) w wyniku ruchu względnego obserwatora i obrazu. Kontur przedmiotu znajduje się w miejscu, w którym  $f(x) = 0$ . Na rysunkach 10b i 10c pokazane są pochodne funkcji  $f(x)$  po czasie wywołane ruchem odpowiednio w prawo lub w lewo. Piki tych funkcji występujące dla  $x = 0$  są bezpośrednio związane z prędkością konturu. Obowiązuje równanie:

$$\frac{df}{dt} = - \frac{df}{dx} \cdot \frac{dx}{dt} = - \frac{df}{dx} \cdot v \quad (38)$$

Stąd prędkość  $v$  konturu dana jest wzorem:

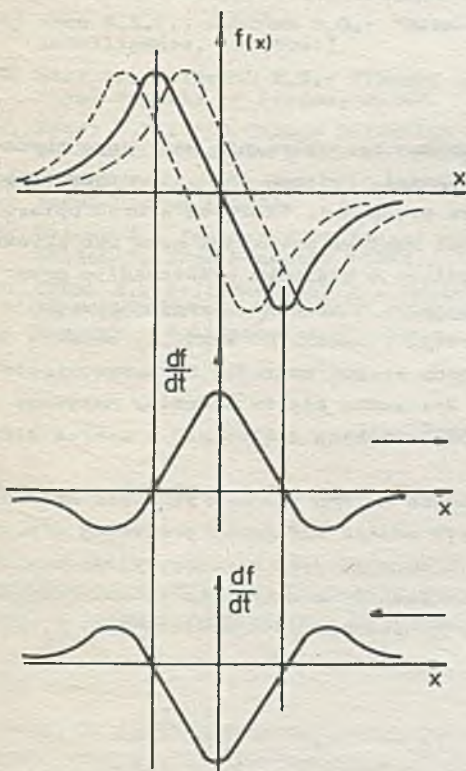
$$v = - \left. \frac{\frac{df}{dt}}{\frac{df}{dx}} \right|_{x=0} \quad (39)$$

$\left. \frac{df}{dt} \right|_{x=0}$  jest wysokością piku na rys. 10b lub 10c.  
 $\left. \frac{df}{dx} \right|_{x=0}$  jest tangensem kąta nachylenia stycznej do krzywej z rys. 10a dla  $x = 0$ .

W przypadku obrazu dwuwymiarowego prędkość w kierunku prostopadłym do konturu może być wyznaczona ze wzoru:

$$v = - \frac{\frac{\partial f(x,y)}{\partial t}}{|\nabla f(x,y)|} \quad (40)$$

Wymagana jest więc znajomość gradientu funkcji  $f(x,y)$  oraz jej pochodnej po czasie. Sposób liczenia gradientu sygnalizowany był w rozdziale 2. Pochodną  $f(x,y)$  po czasie można aproksymować ilorazem:



Rys. 10. Zmiana funkcji  $f(x)$  w czasie wywołana ruchem w prawo lub w lewo

Fig. 10. Method of determining velocity for contour's points

$$\frac{\partial f(x,y)}{\partial t} \approx \frac{\Delta f(x,y)}{\Delta t} = \frac{f_1(x,y) - f_2(x,y)}{\Delta t} \quad (41)$$

$f_1$  oznacza tu funkcję przesuniętą, a  $f_2$  - funkcję pierwotną.

Jeśli  $\frac{\partial f(x,y)}{\partial t}$  jest aproksymowana tak jak we wzorze (41), to istnieje ograniczenie na wielkość przesunięcia obrazu, przy którym aproksymacja nie wprowadza dużych błędów. Ograniczenie to wynika z kształtu funkcji  $f$ . Dla przypadku jednowymiarowego (funkcja  $F$  przedstawiona jest na rys.10) prawidłową aproksymację uzyskuje się, gdy obraz przesunie się nie więcej niż  $\frac{1}{2}$  odległości między pikami funkcji. Odległość między maksimum i minimum funkcji  $f$  zależna jest od kształtu funkcji poziomej szarości oraz od współczynnika  $a$  w równaniu krzywej Gaussa (równanie 19). Dla obrazów o stromej funkcji szarości rozszerzenie zakresu liniowego funkcji  $f$  można uzyskać przez dobranie mniejszej wartości  $a$ .

#### 4. Zakończenie

W pracy przedstawiono pewne algorytmy przetwarzania wstępnego obrazu użyteczne przy wyznaczaniu pola prędkości. Opisane są dwie metody określania składowej prostopadłej wektora prędkości. Pierwsza z nich opiera się na założeniu, że składowa prostopadła wektora prędkości może być aproksymowana przez iloraz przesunięcia konturu w kierunku prostopadłym przez przedział czasu  $\Delta t$  między obserwacjami. W metodzie drugiej przyjmuje się, że prędkość zmian stopnia szarości w czasie na konturze obiektu jest równa w przybliżeniu ilorazowi różnicy stopni szarości na przesuniętych obrazach przez przedział czasu  $\Delta t$ . Wprowadza się tu operator nazywany "laplasjanem z funkcji Gaussa", który zwiększa dokładność i zakres stosowania metody.

Wydaje się, że metoda wykorzystująca zmianę stopnia szarości może być bardziej odpowiednia w przypadku, gdy udział składowej obrotowej w prędkości ruchu wzajemnego obserwatora i obiektu jest znaczny. Wtedy bowiem wektory przesunięcia i prędkości dla punktów konturu nie są równoległe. Metoda ta wymaga jednak znacznie większych nakładów obliczeniowych.

#### Podziękowanie

Autorzy pragną podziękować mgr inż. Andrzejowi Polańskiemu za rzeczowe dyskusje.

## LITERATURA

- [1] Aggarwal J.K., Duda R.O.: "Computer Analysis of Moving Polygonal Images", IEEE Transactions on Computers, c-24 (1975).
- [2] Bow S.T.: "Pattern Recognition", Marcel Decker Inc., New York, (1984)
- [3] Chow W.K., Aggarwal J.K.: "Computer Analysis of Planar Curvilinear Images", IEEE Transaction on Computers, c-26, (1977).
- [4] Dreher B., Sanderson K.I.: "Receptive Field Analysis, Responses to Moving Visual Contours by Single lateral Geniculate Neurons in the Cat", Journal of Physiology, 234 London (1973).
- [5] Fahle M., Poggis T.: "Visual Hyperacuity: Spatiotemporal Interpolation in Human Vision", Proc. of the Royal Society of London B, 213 (1981).
- [6] Frei W., Chen C.C.: "Fast Boundary Detection: A Generalization and a New Algorithm", IEEE Transaction on Computers. c-26 (1976).
- [7] Hildreth E.C.: "The Measurement of Visual Motion", Ph.D. Thesis, Massachusetts Institute of Technology (1983).
- [8] Horn B.K.P., Sohunck B.G.: "Determining Optical Flow" Artificial Intelligence, 17 (1981).
- [9] Marr D., Hildreth E.C.: "Theory of Edge Detection", Proc. of the Royal Society of London, B.207, (1980).
- [10] Nagel H.H.: "On Change Detection and Displacement Vector Estimation in Image Sequences", Pattern Recognition Letters, 1 (1982).
- [11] Rosenfeld A.: "Picture Processing by Computer", Academic Press, New York, London (1969).
- [12] Ullman S.: "Analysis of Visual Motion by Biological and Computer Systems", IEEE Computer (1981).
- [13] Ordys A., Wojciechowski K.: "Metody wyznaczania pola prędkości na podstawie sekwencji obrazów", praca złożona w redakcji Zeszytów Naukowych Politechniki Śląskiej.

Recenzent: Doc. dr hab. Bogdan Wołczak

Wpłynęło do Redakcji 2.01.1987 r.

ИЗБРАННЫЕ АЛГОРИТМЫ ДЛЯ НАЧАЛЬНОГО ПРЕОБРАЗОВАНИЯ  
ОБРАЗА В ПРОБЛЕМЕ ОПРЕДЕЛЕНИЯ ПОЛЯ СКОРОСТИ

Р е з ю м е

В работе даются алгоритмы для начального преобразования образа, которые используются при определении поля скорости. Предполагается, что поле скорости будет рассчитываться по методу минимизации вариации вектора скорости вдоль контура объекта. Оговариваемые алгоритмы связаны с двумя разными методами расчета составной перпендикулярной вектора скорости. Первый метод основан на расчете перемещения контура. Во втором методе используется изменение некоторой функции уровня скорости по контуру.

CHOSEN ALGORITHMS OF THE PRELIMINARY IMAGE PROCESSING  
IN THE PROBLEM OF VELOCITY FIELD ASSIGNMENT

S u m m a r y

In the paper image preprocessing algorithms are presented. They are useful in the assignment of the velocity field. It is assumed that the velocity field will be calculated by minimalization of the variance of the velocity vector along the plant contour. Presented algorithms are connected with two different methods of the calculation of the perpendicular component of the velocity vector. The first method is based on the calculation of the contour displacement. The second method applies the change of the function of the grey level on the contour.