

Algorytmy Laplacian of Gaussian i Canny'ego detekcji krawędzi w procesie analizy satelitarnych obrazów procesów atmosferycznych.

Słowa kluczowe: teledetekcja, filtracja obrazu, segmentacja obrazu, algorytmy wykrywające krawędzie, algorytm Marr-Hildreth, algorytm Canny'ego.

Streszczenie

W artykule skupiono się na algorytmie Canny'ego. Otrzymane wyniki porównano z wynikami filtrowania obrazów satelitarnych za pomocą detektora krawędzi Laplace of Gaussian.

W pracy koncentrowano się na wydobyciu charakterystycznych cech i właściwości obrazów satelitarnych prezentujących zachmurzenie. Mogą one znacząco poprawić ich walory interpretacyjne oraz uwypuklić informacje niezbędne do dalszej analizy i interpretacji.

Celem artykułu jest przedstawienie możliwości wykorzystania badanych filtrów do analizy satelitarnych obrazów procesów atmosferycznych. Praca przedstawia także sposób działania wybranych filtrów w pakiecie Matlab.

Laplacian of Gaussian algorithm and the Canny edge detector in the analysis of satellite images of atmospheric processes.

Keywords: remote sensing, image filtering, image segmentation, edge detection algorithms, the Marr-Hildreth edge detector, the Canny edge detector.

Abstract

The article focuses on the Canny edge detector. Achieved results were compared with the results of filtering of satellite images by Laplacian of Gaussian detector.

The study focused on the extraction of characteristics and properties of satellite images which show cloudy. This characteristics can significantly improve qualities of interpretation and to highlight the information needed for further analysis and understanding.

The purpose of this article is to presents the possibility of using filters to analyze the studied satellite images of atmospheric processes. The paper presents the way of filters working in the package Matlab.

1. Wprowadzenie.

Implementacja algorytmu opiera się na filtrach Laplace'a oraz Gaussa. Dzięki zastosowaniu dolnoprzepustowego (wygładzającego) filtru Gaussa oraz górnoprzepustowego (wyostrzającego) filtru Laplace'a otrzymujemy obraz z wyraźnymi krawędziami.

Dodatkowo w ramach przygotowania obrazu do przetwarzania zastosowano segmentację. Poprzez progowanie wyodrębniono obszary należące do tła oraz do obiektu. Zważywszy na charakter obrazów, jakimi są satelitarne obrazy procesów atmosferycznych tło nie wnosi istotnych informacji do tego typu analiz. W celu polepszenia walorów interpretacyjnych otrzymanych z filtracji obrazów zastosowano operację morfologiczną – ścieniania.

2. Etapy realizacji zmodyfikowanego algorytmu Laplacian z Gaussa

Pierwszym etapem realizacji algorytmu jest splot obrazu wyjściowego z filtrem Gaussa. Zastosowanie filtru Gaussa powoduje usunięcie z obrazu szumu. Operacja ta jest wymagana ze względu na właściwości stosowanego w kolejnym etapie filtru Laplace'a, który powoduje wzmocnienie szumu jest to przedstawione m. in w [2].

Funkcja Gaussa wykorzystana w algorytmie ma następującą postać:

$$G_1(x, y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (1)$$

Ostatecznie do obliczenia nowych wartości pikseli w programie Matlab wykorzystywana jest funkcja:

$$G(x, y) = \frac{G_1(x, y)}{\sum_x \sum_y G_1} \quad (2)$$

W tym miejscu algorytmu podajemy parametr σ – odchylenie standardowe charakteryzujące rozkład Gaussa oraz wielkość maski h . Dla celów tej pracy zastosowano $\sigma = 0.5$ oraz $h=5$ Parametr σ jest proporcjonalny do rozmiaru obszaru na którym działa filtr [3].

Maska filtru (1) wykorzystana do przetworzenia obrazu ma następującą postać

$$\begin{matrix}
0.0000 & 0.0000 & 0.0002 & 0.0000 & 0.0000 \\
0.0000 & 0.0113 & 0.0837 & 0.0113 & 0.0000 \\
0.0002 & 0.0837 & 0.6187 & 0.0837 & 0.0002 \\
0.0000 & 0.0113 & 0.0837 & 0.0113 & 0.0000 \\
0.0000 & 0.0000 & 0.0002 & 0.0000 & 0.0000
\end{matrix} \quad (3)$$



Rys. 1. Zdjęcie wyjściowe



Rys.2. Obraz wejściowy po binaryzacji i zastosowaniu filtru Gaussa dla $\sigma=0.5$ oraz $h=5$.

Kolejnym etapem realizacji algorytmu jest zastosowanie maski Laplace'a. Obrazem wyjściowym jest obraz po filtracji filtrem Gaussa Rys.2. Program Matlab maskę Laplace'a tworzy wykorzystując operator Laplace'a:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (4)$$

Maska filtru generowana jest w programie po podaniu parametru α zgodnie z poniższym wzorem:

$$\nabla^2 = \frac{4}{(\alpha+1)} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix} \quad (5)$$

Przy czym α należy do przedziału $\langle 0.0; 1.0 \rangle$. Dla $\alpha = 0.5$ maska przedstawia się następująco:

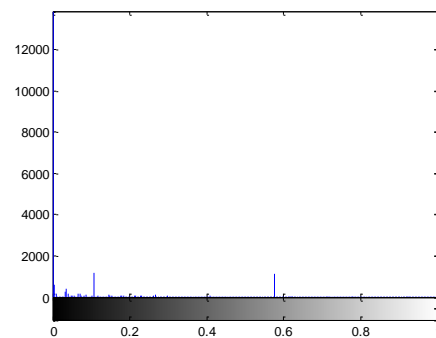
$$\nabla^2 = \begin{bmatrix} 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & -2.6667 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 \end{bmatrix} \quad (6)$$

Rezultaty działania maski (6) na obraz przedstawia Rys. 4.

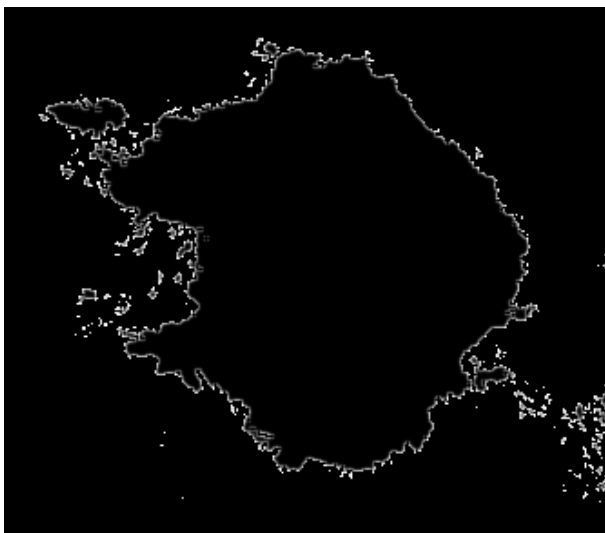
Następnym etapem realizacji algorytmu było znalezienie miejsc zerowych (*zero crossings*). W tym celu przefiltrowano obraz maską Laplace'a w postaci:

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (7)$$

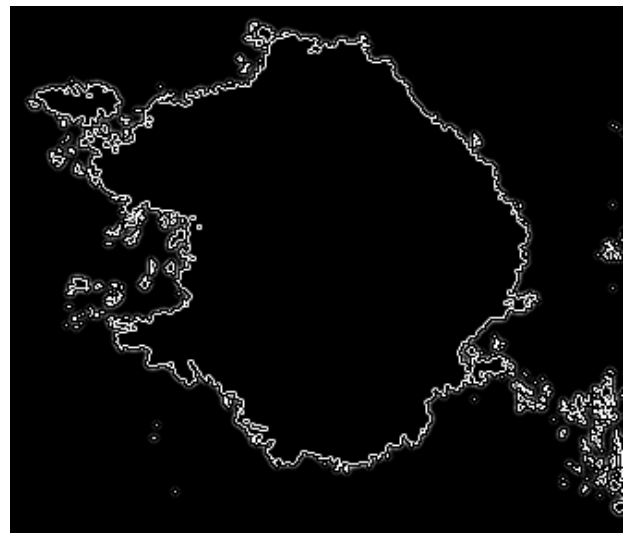
Działanie filtru ilustruje Rys. 5 oraz 6. Powtórne zastosowanie maski Laplace'a spowodowało otrzymanie podwójnej krawędzi. W celu uzyskania obrazu o właściwym stopniu szczegółowości zastosowano binaryzację. Próg binaryzacji dobrano na podstawie histogramu Rys 3.



Rys. 3. Histogram obrazu z Rys. 4.



Rys.4. Obraz z Rys.2. po filtracji filtrem Laplace'a (6) z $\alpha = 0.5$.



Rys. 5. Identyfikacja przejść przez zero w obrazie uzyskanym po filtracji filtrem Laplace'a.



Rys. 6. Obraz z Rys. 5. w powiększeniu. Widoczna podwójna krawędź.



Rys. 7. Obraz z Rys. 5. po zastosowaniu binaryzacji.

W zaprezentowanym tutaj eksperymencie pozyskano etapy pośrednie badanego algorytmu. Dokonano również modyfikacji algorytmu LoG w celu uzyskania jak najlepszych rezultatów dla obrazów satelitarnych prezentujących zachmurzenie.

Uzyskane tą drogą wyniki różnią się od uzyskanych za pomocą filtracji filtrem LoG. Przyczyną różnic jest wykorzystanie w filtrze LoG czynnika normalizacji (10). Rozłożenie algorytmu na etapy pośrednie stworzyło również większe możliwości modyfikacji jego działania poprzez możliwość definicji większej liczby parametrów.

2.1. Etapy realizacji algorytmu Laplacian z Gaussa w pakiecie Matlab

Filtr LoG jest zdefiniowany w pakiecie Matlab, jako jeden ze specjalistycznych filtrów wykrywających krawędzie [4]. Generowanie filtru oparte jest o funkcję (1) oraz (4). Przy detekcji krawędzi wykorzystuje się właściwość operatora Laplace'a polegającą na zmianie znaku drugiej pochodnej dla punktu środkowego krawędzi. Filtr Gaussa stosuje się w celu stłumienia składowych dużych częstotliwości.

Filtracja Laplace'a w połączeniu z filtrem Gaussa daje w wyniku:

$$\nabla^2 G_1(x, y) = \frac{\partial^2 G_1(x, y)}{\partial x^2} + \frac{\partial^2 G_1(x, y)}{\partial y^2} \quad (8)$$

$$\nabla^2 G_1(x, y) = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \quad (9)$$

Wyrażenie (9) nazywane jest Laplasjanem gaussowskim (*Laplacian of a Gaussian*, LoG). Funkcja (9) często uzupełniana jest o współczynnik normalizacji:

$$\frac{1}{2\pi\sigma^2} \quad (10)$$

Współczynnik (10) odpowiada za normalizację elementów maski, których suma wynosi 0 [2].

W programie Matlab tworzenie maski filtru odbywa się w sposób automatyczny po podaniu parametru σ oraz rozmiaru maski h zgodnie z wyrażeniem:

$$\nabla^2 G_1(x, y)_M = \frac{(x^2 + y^2 - 2\sigma^2)G_1(x, y)}{2\pi\sigma^6 \sum_x \sum_y G_1} \quad (11)$$

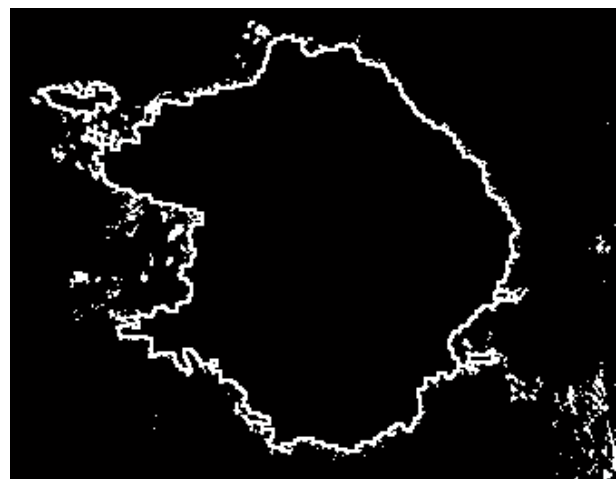
Maska filtru stanowiąca dyskretną aproksymację funkcji LoG dla $h=[5 \ 5]$ i $\sigma=0.5$:

$$\begin{array}{ccccc} 0.0448 & 0.0468 & 0.0564 & 0.0468 & 0.0448 \\ 0.0468 & 0.3167 & 0.7146 & 0.3167 & 0.0468 \\ 0.0564 & 0.7146 & -4.9048 & 0.7146 & 0.0564 \\ 0.0468 & 0.3167 & 0.7146 & 0.3167 & 0.0468 \\ 0.0448 & 0.0468 & 0.0564 & 0.0468 & 0.0448 \end{array} \quad (12)$$

Działanie filtra LoG ilustruje Rys. 8 i 9.



Rys. 8. Obraz poddany działaniu filtra LoG dla $h=[5 \ 5]$ i $\sigma=0.5$ (12).



Rys. 9. Obraz poddany działaniu filtra LoG dla $h=[5 \ 5]$ i $\sigma=0.1$.

3. Algorytm Canne'go detekcji krawędzi w procesie analizy satelitarnych obrazów zachmurzenia

Detekcja krawędzi algorytmem Canne'go ma na celu:

1. Dobrą detekcję – wszystkie krawędzie powinny być znalezione bez fałszywych odpowiedzi.

2. Dobrą lokalizację - wykryte krawędzie muszą być jak najbliżej istniejących krawędzi.
3. Wyraźną odpowiedź - detektor powinien zwracać tylko jeden punkt dla każdego istniejącego punktu krawędzi.

Algorytm stanowi złożoną procedurę detekcji krawędzi. Działanie algorytmu Canne'go można przedstawić w czterech etapach. Są to:

1. Wygładzenie obrazu za pomocą filtru gaussowskiego.
2. Obliczenie w każdym punkcie obrazu wartości oraz kierunku gradientu.
3. Zastosowanie tłumienia niemaksymalnych pikseli w celu zmniejszenia grubości krawędzi.
4. Przeprowadzenie progowania z histerezą w celu minimalizacji występowania fałszywych krawędzi.

Proces przebiegu algorytmu dokładnie opisany jest m. in. w [1,3].

3.1. Etapy realizacji algorytmu Canne'go w pakiecie MATLAB na przykładzie satelitarnych zdjęć procesów atmosferycznych

Pierwszym etapem procesu przetwarzania obrazu w programie MATLAB za pomocą algorytmu Canne'go jest redukcja szumów za pomocą filtru Gaussa (1).

Podajemy tu pierwszy parametr σ – odchylenie standardowe charakteryzujące rozkład Gaussa. Rozmiar maski generowany jest automatycznie na podstawie wartości σ . Do wygenerowania rozmiaru maski program Matlab wykorzystuje następujące wyrażenie:

$$G(pw) = e^{\frac{-pw^2}{2\sigma^2}} > 0.0001 \quad (13)$$

Wyrażenie to jest funkcją Gaussa jednej zmiennej z dodatkowym warunkiem. Zdefiniowana w programie funkcja działa na bazie wyrażenia (13), w taki sposób, że zwraca ostatni indeks elementu tablicy spełniającego wyrażenie. Wartość pw to wygenerowana w programie trzydziestoelementowa jednowierszowa macierz – możliwa szerokość maski. Jeśli użytkownik poda zbyt małą wartość σ , program automatycznie przyjmie, że indeks wynosi 1. Następnie program generuje jednowierszową macierz o wartościach od minus indeks do indeks co jeden. Zgodnie z tym algorytmem dla indeksu 1 otrzymujemy -1, 0, 1 -szerokość maski wynosi 3.

Jednowymiarowa maska filtru tworzona jest zgodnie ze wzorem:

$$G(t) = e^{\frac{-t^2}{2\pi\sigma^2}} \quad (14)$$

Gdzie t – jest jednowierszową macierzą od minus indeks do indeks, dla $\sigma = 0.5$

$$t = [-2 \quad -1 \quad 0 \quad 1 \quad 2].$$

Maska filtru stanowiąca aproksymację (14) dla $\sigma = 0.5$ wygląda następująco:

$$G(t) = [0.0002 \quad 0.0862 \quad 0.6366 \quad 0.0862 \quad 0.0002] \quad (15)$$

Kolejnym etapem realizacji algorytmu jest znalezienie kierunkowej pochodnej dwuwymiarowej funkcji Gaussa (wzdłuż osi X). Pochodną wzdłuż osi Y uzyskamy dokonując transpozycji otrzymanego wyniku.

W pakiecie MATLAB dwuwymiarowa funkcja Gaussa przedstawia się wzorem (16).

$$G(x, y) = -xe^{\frac{-(x^2+y^2)}{2\sigma^2}} \frac{1}{\pi\sigma^2} \quad (16)$$

Gdzie x i y to odpowiednio:

$$\begin{array}{r}
 \begin{array}{ccccc}
 -2 & -1 & 0 & 1 & 2 \\
 -2 & -1 & 0 & 1 & 2 \\
 x = & -2 & -1 & 0 & 1 & 2 \\
 -2 & -1 & 0 & 1 & 2 \\
 -2 & -1 & 0 & 1 & 2
 \end{array}
 &
 \begin{array}{ccccc}
 -2 & -2 & -2 & -2 & -2 \\
 -1 & -1 & -1 & -1 & -1 \\
 y = & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 1 & 1 & 1 \\
 2 & 2 & 2 & 2 & 2
 \end{array}
 \end{array}$$

Maska filtru będąca aproksymacją (16):

$$\begin{array}{r}
 \begin{array}{ccccc}
 0.0000 & 0.0001 & 0 & -0.0001 & -0.0000 \\
 0.0001 & 0.0233 & 0 & -0.0233 & -0.0001 \\
 G = & 0.0009 & 0.1723 & 0 & -0.1723 & -0.0009 \\
 0.0001 & 0.0233 & 0 & -0.0233 & -0.0001 \\
 0.0000 & 0.0001 & 0 & -0.0001 & -0.0000
 \end{array}
 \end{array} \quad (17)$$

Kolejnym etapem algorytmu jest wykonanie splotu obrazu z dwuwymiarowym filtrem Gaussa, a następnie z pochodną funkcji Gaussa. Operację tę można zastąpić splotem z jednowymiarowym filtrem Gaussa (15) wzdłuż wierszy, a następnie wzdłuż kolumn. Ze względu na symetryczność filtru Gaussa otrzymane wyniki będą poprawne.

Rezultat przedstawiony jest na Rys. 10.



Rys. 10. Obraz oryginalny, obraz poddany binaryzacji i obraz wygładzony filtrem Gaussa z parametrem $\sigma = 0.5$.

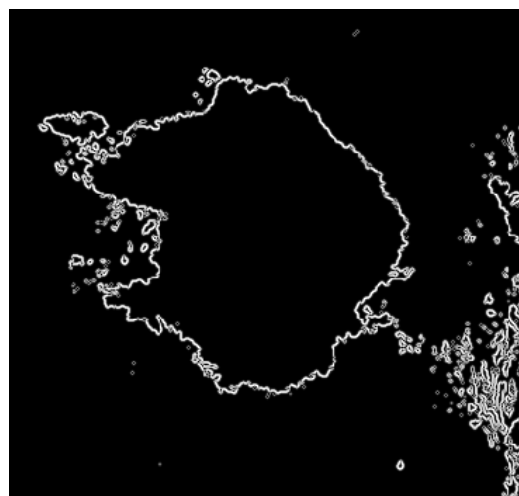
W wyniku splotu obrazu w kierunku wierszy i kolumn otrzymujemy składowe gradientu w dwóch prostopadłych kierunkach. Obliczany jest moduł gradientu obrazu zgodnie ze wzorem:

$$Mag = \sqrt{(ax^2 + ay^2)} \quad (18)$$

Gdzie: ax , ay – macierze obrazu uzyskane po operacji splotu z dwuwymiarowym filtrem Gaussa odpowiednio w kierunku osi x i y .



Rys. 11. Moduł gradientu obrazu. Wynik dla obrazu binarnego $\sigma = 0.5$.



Rys. 12. Obraz po normalizacji modułu gradientu $\sigma = 0.5$.

Przed rozpoczęciem procesu tłumienia niemaksymalnego należy podać dwa parametry, które będą wykorzystane do ostatnich dwóch etapów algorytmu Canne'go.

Przed realizacją progowania z histerezą wykonywane jest tłumienie niemaksymalne (Non-Maximal suppression). Proces ten ma na celu znalezienie centra krawędzi i odrzucenie szumu. Pikselem krawędzi pozostanie ten, który będzie miał najwyższą wartość odpowiedzi krawędzi spośród pikseli sąsiednich w tym samym

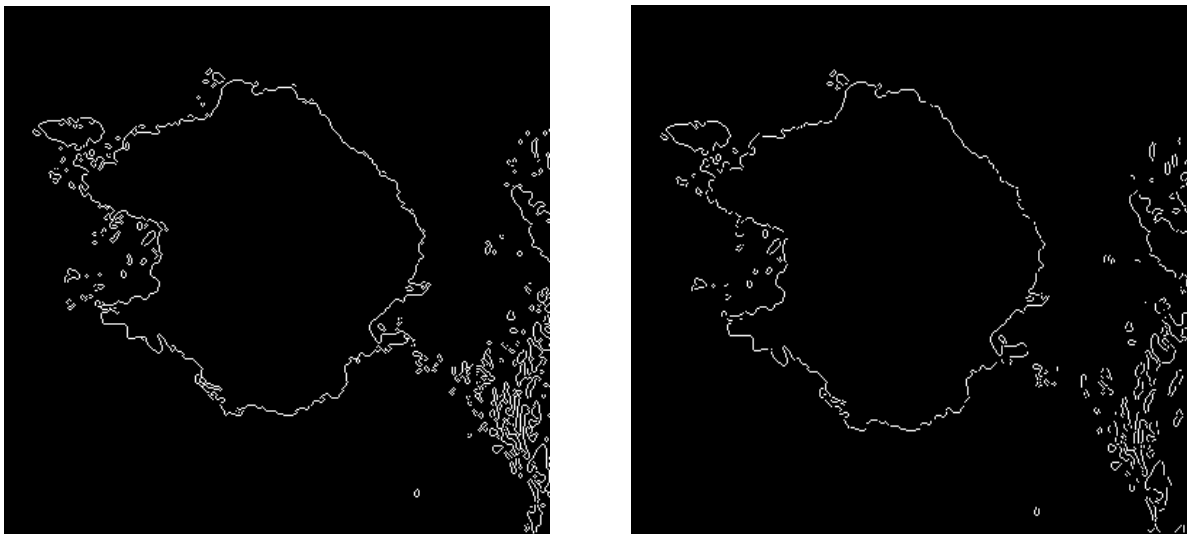
kierunku. Eliminacja nie maksymalnych wartości natężeń pikseli powoduje uzyskanie krawędzi o grubości jednego piksela, co stanowi cel algorytmu Canne'go.

Końcowym etapem procesu jest ustalenie poziomu szczegółowości wykrytych krawędzi. Implementacja progowania z histerezą wymaga zastosowania dwóch parametrów. Są to próg wysoki T_H oraz niski T_L . Istota tego procesu polega na tym, że wszystkie piksele, których wartości są wyższe od wysokiego progu przypisywane są jako pozytyw. Piksele o wartościach znajdujących się pomiędzy progami są również przypisywane jako pozytyw. Pozostałe piksele są zadeklarowane jako negatyw [3].

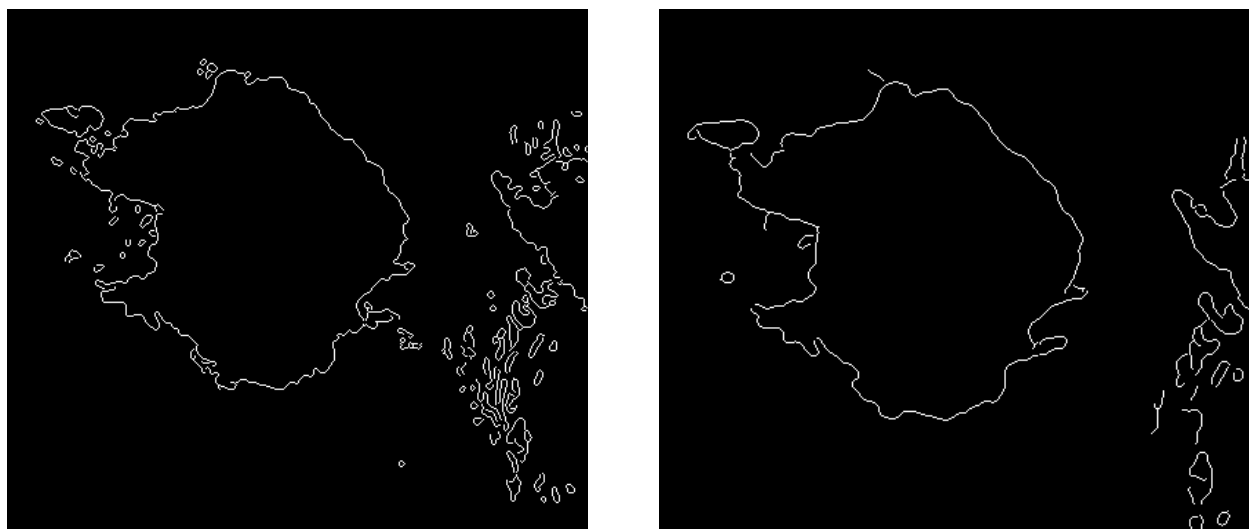
W programie MATLAB użytkownik podaje wektor składający się z wartości progów. W przypadku, gdy został zdefiniowany skalar program przyjmuje tą wartość jako próg wysoki T_H , na podstawie którego obliczany jest próg niski zgodnie ze wzorem:

$$T_L = 0.4 * T_H \quad (19)$$

Rys. 13. oraz 14. prezentują działanie pełnego algorytmu Canne'go detekcji krawędzi zaimplementowanego w pakiecie MATLAB oraz wpływ parametrów filtracji na wynik końcowy.



Rys. 13. Wpływ wyboru progu na wynik działania algorytmu. Od lewej: obraz z progiem $T_L = 0.4$, $T_H = 0.6$, oraz $T_L = 0.8$, $T_H = 0.9$ dla parametru $\sigma = 0.5$.



Rys. 14. Wpływ wyboru parametru σ na wynik działania algorytmu. Od lewej: $T_L = 0.4$, $T_H = 0.6$, $\sigma = 1$, $T_L = 0.4$, $T_H = 0.6$, $\sigma = 2$.

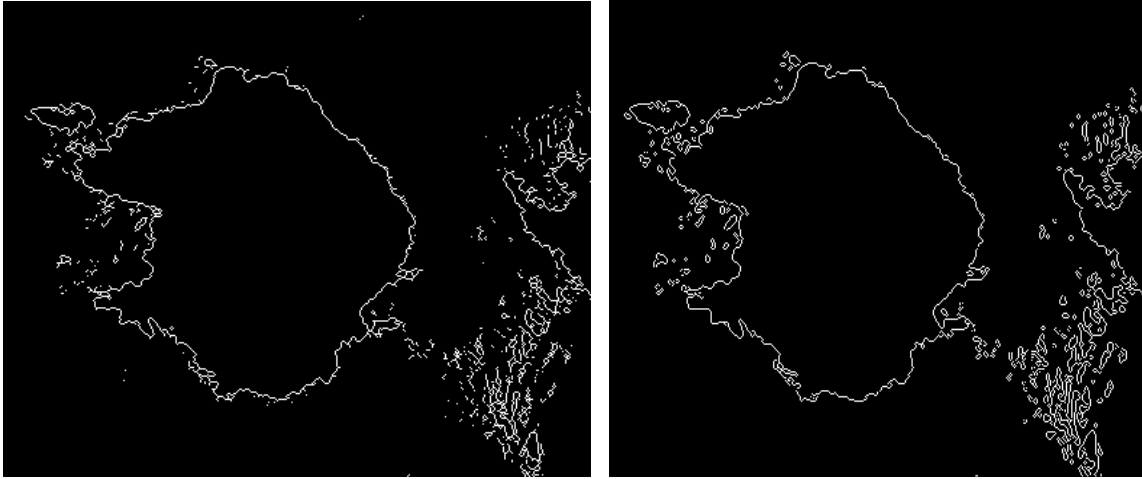
Wyniki otrzymane z przeprowadzonych badań pozwalają stwierdzić, że im większa wartość odchylenia standardowego funkcji Gaussa tym bardziej radykalne działanie filtru. Kluczowe znaczenie ma także prawidłowy dobór progów w procesie progowania z histerezą. Źle zdefiniowane wartości progów spowodują uzyskanie nieciągłych krawędzi.

4. Porównanie wyników i wnioski końcowe

Jeden detektor krawędzi może wykrywać nie tylko krawędzie ale także szumy zwłaszcza w przypadku niewłaściwie dobranych parametrów filtracji, wspomina o tym [2], dlatego w pracy porównano wyniki otrzymane z obu metod.

Poddając analizie wizualnej otrzymane rezultaty można stwierdzić, że w przypadku zastosowania w obu metodach tej samej wartości σ oraz dodatkowo zdefiniowania tego samego rozmiaru maski wyniki są bardzo zbliżone. Dodatkowo zastosowano operację morfologiczną ścieniania dla obrazu filtrowanego algorytmem LoG. Pozwoliło to na łatwiejsze porównanie wyników otrzymanych z obu metod wykrywania krawędzi

Rys. 15.



Rys. 15. Porównanie obrazu filtrowanego filtrem LoG i za pomocą algorytmu Cannego. Od lewej – LoG - $\sigma = 0,5$, $H=[5,5]$ po operacji ścieniania, Canny - $T_L = 0.4$, $T_H = 0.6$, $\sigma = 0.5$.

Porównując rezultaty pod względem jakości detekcji algorytm Canne'go pozwolił uzyskać lepsze wyniki. Detektor krawędzi Canne'go jest bardziej złożony niż filtr LoG. Dzięki zastosowaniu tłumienia niemaksymalnego otrzymane krawędzie są grubości 1 piksela, a progowanie z histerezą redukuje prawdopodobieństwo wykrycia szumu.

Literatura

1. Gonzalez R. C. Woods R. E.: *Digital Image Processing*, Upper Saddle River, NJ: Pearson Education, 2008.
2. Jähne B.: *Digital Image Processing*, Springer-Verlag, Berlin 2002.
3. Malina W., Smiatacz M.: *Metody cyfrowego przetwarzania obrazów*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2005.
4. Wróbel Z., Koprowski R.: *Praktyka przetwarzania obrazów z zadaniami w programie Matlab.*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2008.