Tomasz Strzałka
gr. 26A

Analiza obrazów z wykorzystaniem falek do detekcji uszkodzeń konstrukcji

Spis treści

Wprowadzenie

Uszkodzenia konstrukcji można zdefiniować jako zmianę stanu materiału, która powoduje chwilowe lub stałe zaburzenie funkcjonowanie konstrukcji i może doprowadzić do zniszczenia elementu bądź w skrajnych sytuacjach całej konstrukcji. Defekty mają postać delaminacji (rozwarstwienia, utraty spójności poszczególnych warstw składowych), rys, miejscowego zniszczenia materiału z powodu korozji lub zmęczenia.

Ocena stanu konstrukcji oraz wczesne wykrycie uszkodzeń jest kluczowym elementem utrzymania konstrukcji, od tego zależy bezpieczeństwo oraz trwałość konstrukcji. Rosnącą popularnością cieszą się badania nieniszczące, które dostarczają informacji o nieciągłości w obiekcie bez naruszenia ciągłości struktury oraz powodowania zmian i wpływania na jego właściwości użytkowe. Jednym z rodzajów badań nieniszczących są badania wizualne, wykorzystywane zwykle jako badania wstępne konstrukcji. Badania te pozwalają na wykryciu nieciągłości powierzchniowych (pęknięć). Często wykonywane są nieuzbrojonym okiem i ich dokładność zależy od doświadczenia, wiedzy oraz dobrego oka osoby przeprowadzającej badania, lecz można także wykorzystać do tego celu komputerową analizę i przetwarzanie obrazu.

Do wykrywania nieciągłości w komputerowej analizie obrazu służą metody detekcji krawędzi (edge detection) takie jak algorytm Canny`ego czy algorytmy korzystające z operatorów Prewitta, Sobela czy filtru Robertsa. Stosunkowo młodą metodą pozwalającą na detekcję krawędzi, jest mająca podstawy w matematycznej teorii analizy sygnałów transformacja falkowa. Dzięki możliwości wielorozdzielczej dekompozycji pozwala na wydobywanie nawet małych lokalnych zaburzeń sygnału. Możliwe jest analizowanie sygnału na wielu poziomach szczegółowości .Jedną z zalet tej metody jest fakt, że dyskretna transformacja falkowa umożliwia wykrycie defektu jedynie na podstawie analizy obrazu uszkodzonej konstrukcji, bez konieczności porównywania z konstrukcją nieuszkodzoną.

Podstawy analizy falkowej

Definicja falki:

Zgodnie z [2] funkcja $\psi(t) \in L^2(\mathbb{R})$ jest dopuszczalną falką podstawową, jeśli:

$$C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{(\omega)} d\omega < \infty \tag{1.1}$$

przy czym $\Psi(\omega)$ jest transformatą Fouriera funkcji $\psi(\omega)$.

Warunek ten można interpretować jako wymaganie, by przy ω dążącym do nieskończoności, $|\Psi(\omega)|^2$ dążyło do zera szybciej niż $1/\omega$.

Z warunku tego, przy założonej całkowalności z kwadratem wynika, że

$$\int \psi(t)\, dt = 0 \quad , \tag{1.2}$$

co jest równoznaczne z

$$\Psi(0) = 0 \tag{1.3}$$

Biorąc pod uwagę, że falka podstawowa jest dobrze skoncentrowana w czasie i częstotliwości, powyższe równości oznaczają, że funkcja $\psi(t)$ ma co najmniej kilka oscylacji. Jest to więc krótka fala i stąd pochodzi nazwa falka.

Funkcją, która spełnia podany warunek dopuszczalności falki jest funkcja falki $\psi(t)$:

$$\psi_{mn}(t) = a_0^{(-m/2)} \psi(a_0^{(-m)} t - n b_0) \quad , \tag{1.4}$$

gdzie m i n są liczbami całkowitymi, ustalone parametry a_0 oraz b_0 są rzeczywiste i dodatnie, a a_0 jest dodatkowo większe od jedności. Ponieważ $a_0 > 1$, więc $m < 0$ oznacza oscylacje falki o mniejszej częstotliwości, czyli ta sama liczba oscylacji zajmuje szerszy przedział czasowy niż w przypadku falki, dla której $m > 0$ .Zmieniając m uzyskujemy zatem zmianę upakowania oscylacji – uzyskujemy falki o szerszych lub węższych zakresach częstotliwości. Dla ustalonego m falki ψ_{mn} są przesunięciami o $n a_0^m b_0$ falki podstawowej:

$$\psi_{m0} = a_0^{-m/2} \psi \frac{t}{a_0^m} \tag{1.5}$$

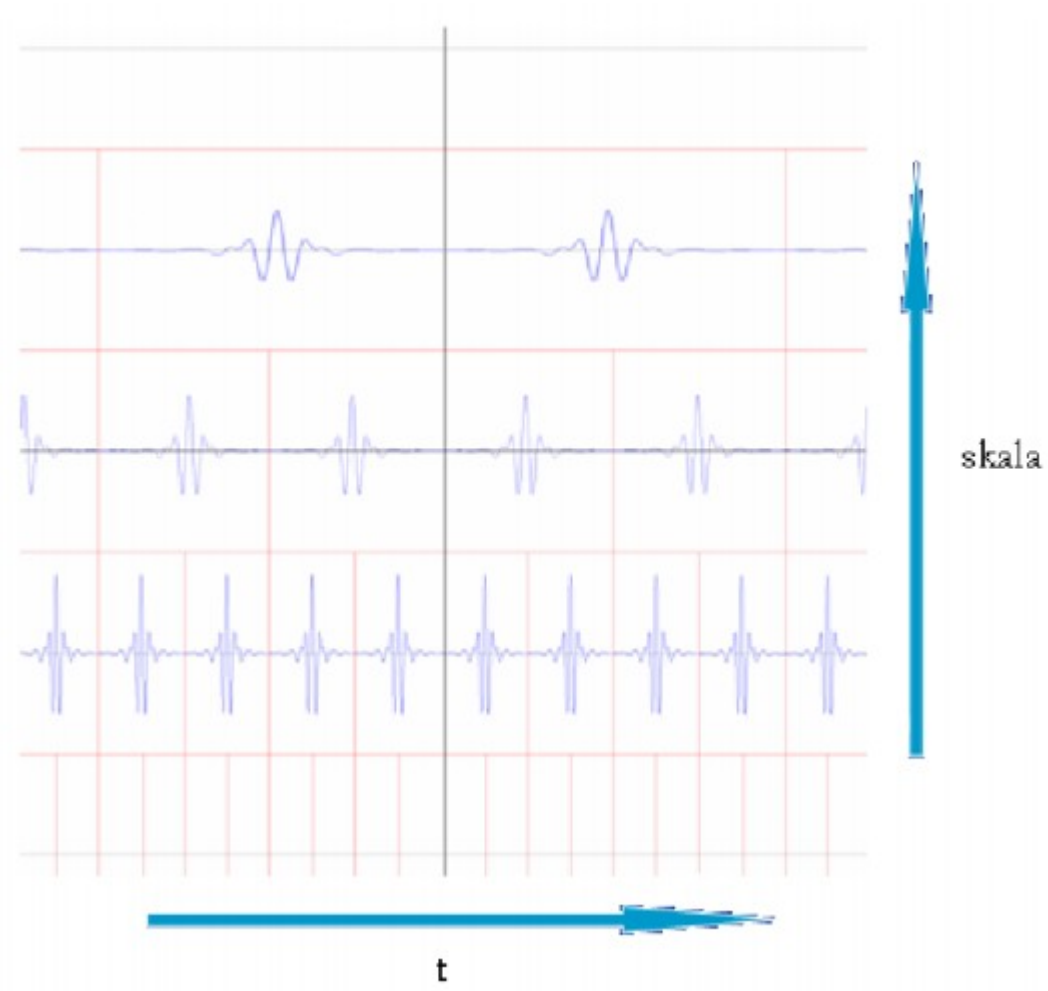
czyli falki są przesuwane o przedział czasowy proporcjonalny do ich szerokości.

Rodzina funkcji falkowych

Rodzinę funkcji falkowych stanowi zbiór funkcji powstałych w wyniku operacji skalowania i przesuwania funkcji podstawowej (1.4). Parametr m jest parametrem skali, a n parametrem przesunięcia. Zmiana parametru m wpływa na kształt falki podstawowej – wraz ze wzrostem tego parametru falka zostaje rozciągnięta w czasie. W przypadku zmiany parametru n dochodzi jedynie do przesunięcia falki na osi t.

Dla parametru $m \to \infty$ szerokość nośnika (przedziału, w którym sygnał elementarny jest niezerowy) rośnie do nieskończoności, co powoduje, że szerokość jej widma maleje do 0 oraz przesuwa się ono w kierunku niższych częstotliwości – dla $m \to \infty$ uzyskujemy nieskończenie ostrą lokalizację względem częstotliwości (widmo zostało zredukowane do jednego piku) oraz bardzo niedokładną względem czasu (nośnik zajmuje całą oś czasu). Z kolei dla $m \to -\infty$ szerokość nośnika dąży do 0, szerokość jej widma dąży do nieskończoności – uzyskujemy nieskończenie ostrą lokalizację w czasie (co pozwala na dowolną dokładność lokalizacji nieciągłości) oraz bardzo niedokładną lokalizację względem częstotliwości.

Poniższy rysunek ukazuje rozmieszczenie poszczególnych falek na płaszczyźnie czas, skala.



Rysunek 1: Rozmieszczenie falek na płaszczyźnie czas,skala

Ciągła i dyskretna transformacja falkowa

Jeśli $f(t)\in L^2$ i $\psi(t)$ jest dopuszczalną falką podstawową, to funkcja $f(t)$ może być w pełni zrekonstruowana przy wykorzystaniu jej **ciągłej transformaty falkowej**:

$$Wf(a,b)=(f,\psi_{ab})=\int_{-\infty}^{\infty} f(t)\psi_{ab}(t)'dt \quad , \tag{1.6}$$

przy czym transformata odwrotna ma postać:

$$f(t)=\frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(f,\psi_{ab})\psi_{ab}(t)\frac{da\,db}{a^2} \tag{1.7}$$

przy czym C_ψ opisane jest równaniem (1.1). Równanie (1.7) jest przepisem na rekonstrukcję $f(t)$, gdy znana jest jej transformata falkowa $Wf(a,b)$. Można też równość tą interpretować jako sposób przestawienia $f(t)$ w postaci superpozycji falek $\psi_{ab}(t)$, przy czym współczynniki tej superpozycji są określone przez transformatę falkową funkcji $f(t)$.

Rozważając równanie (1.4) można wybrać $a_0=2$ oraz $b_0=1$. Mamy wówczas do czynienia z diadyczną ortonormalną bazą falkową przestrzeni $L^2(\mathbb{R})$

$$\psi_{mn}(t)=2^{(-m/2)}\psi(2^{-m}t-n)_{(m,n)\in\mathbb{Z}^2} \tag{1.8}$$

przy czym dla ustalonego m mamy stałą skalę 2^m. Zmiana parametru m o 1 oznacza dwukrotną zmianę skali. Równanie (1.8) można zapisać w postaci:

$$\psi_{mn}(t)=2^{(-m/2)}\psi(2^{(-m)}(t-2^m n)) \tag{1.9}$$

i wyróżniając $\psi_{m0}(t)$ jako falkę podstawową dla ustalonego poziomu skali m_0 otrzymujemy:

$$\psi_{mn}=\psi_{m0}(t-2^m n) \quad . \tag{1.10}$$

Falka $\psi_{m0}(t)$ przesuwana jest w czasie na poziomej ustalonej skali m z krokiem równym 2^m. Kolejne potęgi dwójki wyznaczają próbkowanie na osi skali.

Dyskretna analiza falkowa sygnału $f(t)$ sprowadza się do wyznaczenia jego dyskretnych transformat falkowych, które wyznaczamy jako iloczyny skalarne funkcji f i ciągu funkcji ψ_{mn}. Iloczyny te nazywane są współczynnikami falkowymi.

$$(f,\psi_{mn})=\int f(t)\psi_{mn}(t)'dt \tag{1.11}$$

Dyskretna transformata odwrotna definiowana jest równością:

$$f(t)=\sum_{m,n}(f,\psi_{mn})\psi_{mn}=\sum_m\sum_n d_m[n]\psi_{mn} \quad . \tag{1.12}$$

Współczynniki falkowe $d_m[n]=(f,\psi_{mn})$ reprezentują wspólne cechy sygnału i falki – parametry m oraz n zapewniają dostęp do określonych cech sygnału. Parametr n umożliwia zlokalizowanie chwili, w której chcemy dokonać analizy sygnału, a parametr m umożliwia wybranie poziomu skali czy też zakresu częstotliwości, w którym chcemy zbadać widmo częstotliwościowe sygnału.

Analiza wielorozdzielcza

W przypadku, gdy chcemy wykorzystać falki do analizy wielorozdzielczej, musimy posiadać unikatowy zbiór dwóch funkcji – funkcji falkowej (zwanej falką matką) oraz funkcji skalującej (zwanej córką). Tym co odróżnia funkcję skalującą od falki jest niezerowa wartość średnia. Funkcja skalująca opisana jest wzorem:

$$\phi_{m,n}(t)=2^{m/2}\phi(2^m t-n) \tag{1.13}$$

Analiza wielorozdzielcza (MRA – Multiresolution Analysis) jest ściśle związana z falkami. Aby pokazać algorytm działania należy wprowadzić pojęcie ciągów przestrzeni funkcyjnych. W tych przestrzeniach dane sygnały cyfrowe wyrażona są za pomocą różnej liczby funkcji bazowych oraz aproksymowane z różną dokładnością. Przestrzenie funkcyjne oznacza się symbolem V_m , gdzie $m\in\mathbb{Z}$.

Przestrzenie te spełniają postulaty Mallat'a i Meyer'a:

- kolejne przestrzenie zawierają się w sobie: $V_{-1}\subset V_0\subset V_1$
- przestrzeń na (m+1)-wszym poziomie rozdzielczości składa się z przestrzeni na m-tym poziomie rozdzielczości oraz dopełnienia W_m
- nie istnieje funkcja, która należy do wszystkich przestrzeni oraz domknięcie wszystkich przestrzeni daje przestrzeń funkcji o ograniczonej energii
- wszystkie przestrzenie są skalowaną wersją jednej przestrzeni i przemnożenie argumentu funkcji reprezentującej sygnał przez 2 powoduje przemieszczenie go do kolejnej przestrzeni (do następnego poziomu rozdzielczości)
- operacja translacji nie powoduje przeniesienia funkcji reprezentującej sygnał do innej przestrzeni

Dowolny sygnał $x(t)$ można przedstawić jako sumę przesuwalnych funkcji bazowych, należących do przestrzeni V_m :

$$x(t)\approx c_{m,n}\phi(2^m t-n) \tag{1.14}$$

Dzięki analizie wielorozdzielczej można przedstawić ten sygnał w przestrzeni V_{m-1} - czyli z dwukrotnie mniejszą rozdzielczością:

$$\begin{aligned} x(t)&\approx\sum_n c_{m-1,n}\phi(2^{m-1}t-n)\\ x(t)&\approx\sum_n c_{m-1,n}\phi(\frac{2^m}{2}t-n)\end{aligned} \tag{1.15}$$

Przy zmianie rozdzielczości z większej na mniejszą (jak w tym przypadku) tracona jest część informacji. Zgodnie z postulatami Mallat'a i Meyer'a utracone szczegóły można zebrać w przestrzeni W_{m-1} . Dlatego też sygnał aproksymowany z rozdzielczością m można przedstawić równoważnie w rozdzielczości $m-1$

$$x(t)\approx\sum_n c_{m,n}\phi(2^m t-n)=\sum_n c_{m-1,n}\phi(2^{m-1}t-n)+\sum_n d_{m-1,n}\phi(2^{m-1}t-n) \tag{1.16}$$

Współczynniki $c_{m,n}$ oraz $d_{m,n}$ są odpowiednio równe:

$$c_{m,n}=\int x(t)\phi'(2^m t-n)dt \tag{1.17}$$

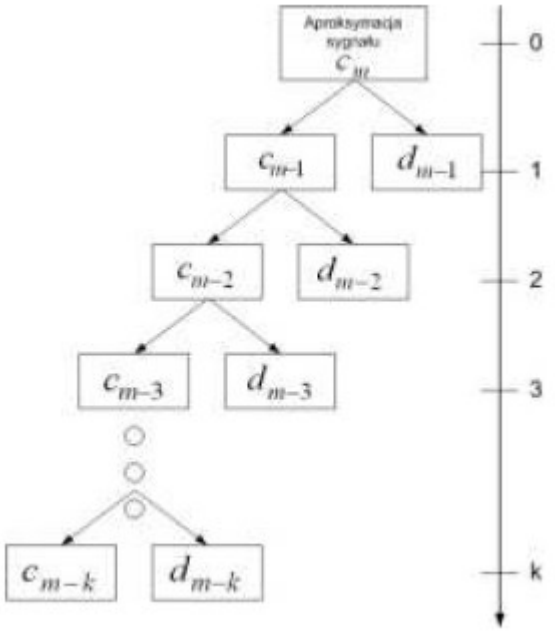
$$d_{m,n}=\int x(t)\psi'(2^m t-n)dt \tag{1.18}$$

gdzie $\psi_{m,n}$ to falki, funkcje bazowe zawierające się w przestrzeni W_m , a $\phi_{m,n}$ to funkcje skalujące.

Zgodnie z równaniem (1.16) sygnał $x(t)$ można przedstawić jako sumę aproksymacji (wyrażenia ze współczynnikami c) oraz detali (wyrażenia ze współczynnikami d). Przejście z rozdzielczości m do $m-1$ nazywamy dekompozycją jednopoziomową. Analogicznie aproksymację na pierwszym poziomie dekompozycji można rozbić na kolejne aproksymacje i detale na drugim poziomie dekompozycji (rozdzielczość $m-2$). Czynność tą można powtarzać aż do osiągnięcia k-tego poziomu dekompozycji. Wówczas sygnał można przedstawić za pomocą wyrażenia:

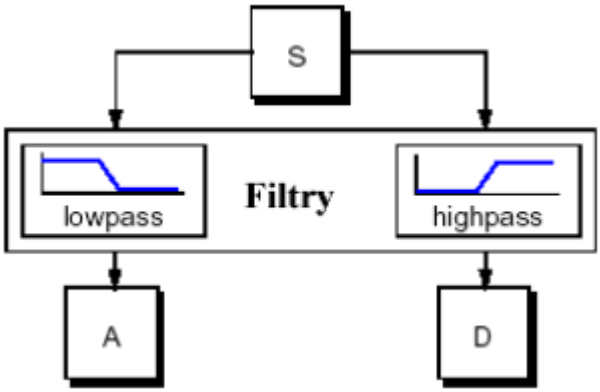
$$x(t)=\sum_n c_{m-k,n}\phi(2^{m-k}t-n)+\sum_{i=0}^{k}\sum_n d_{m-i,n}\psi(2^{m-i}t-n) \tag{1.19}$$

Algorytm ten zwany algorytmem Mallat'a przedstawia poniższy schemat:



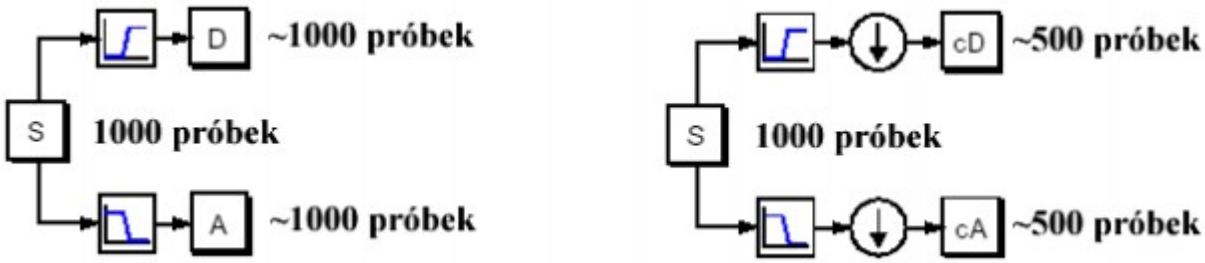
Rysunek 2: Drzewo dekompozycji sygnału

W praktyce nie korzysta się bezpośrednio z wyżej przedstawionych wzorów (1.17) oraz (1.18). Aby uniknąć korzystania z nich wprowadzono specjalnie zaprojektowane filtry. Zauważono, że falki przypominają filtry górnoprzepustowe (H - high-pass filter), natomiast funkcje skalujące są podobne do filtrów dolnoprzepustowych (L - low-pass filter). Oryginalny sygnał przechodzi przez te dwa filtry i w efekcie mamy 2 sygnały wyjściowe. Jeśli operacja ta wykonywana jest na rzeczywistym sygnale cyfrowym to używanych jest dwukrotnie więcej danych niż było na wejściu.



Rysunek 3: Schemat dekompozycji sygnału na składowe

Zakładając, że oryginalny sygnał składał się z tysiąca próbek danych, wtedy aproksymacje i detale również mają po 1000 próbek – czyli łącznie 2000 próbek danych. W celu rozwiązania tego problemu wprowadzono decymację (downsampling) – wyrzucenie co drugiego punktu danych.



Rysunek 4: Ilustracja procedury decymacji sygnału

W praktyce stosuje się zespół czterech filtrów – dwóch do analizy i dwóch do syntezy, powiązanych ze sobą i charakterystycznych dla danej falki. Odwrotna transformata falkowa (rekonstrukcja sygnału) polega na dodawaniu detali i aproksymacji poddanych procesowi upsamplingu (dodaniu zerowej próbki co drugą próbkę sygnału), a następnie przepuszczonych przez filtry odwrotne do filtrów H oraz G.

Falki w analizie obrazu

Obraz można traktować jako funkcję dwóch zmiennych przedstawiającą jasność w punktach o współrzędnych (x, y) . Funkcję dwóch zmiennych można przekształcić w funkcję jednej zmiennej poprzez ustalenie drugiej zmiennej – np. poprzez podstawienie danej wartości $y=y_0$ otrzymując funkcję jednej zmiennej $f(x, y_0)$ Funkcja $f(x, y)$ z podstawieniem $x=x_0$ jest przekrojem obrazu wzdłuż prostej równoległej do osi X,a z podstawieniem $y=y_0$ jest przekrojem obrazu wzdłuż prostej równoległej do osi Y. Przekroje (sygnały jednowymiarowe) są rozwijalne w bazach przesuwnych i złożonych.

Rozważając funkcję dwóch zmiennych $g(x, y)$ oraz jej przekroje $g(x, y_0)$ oraz $g(x_0, y)$, które dają się rozwinąć w tej samej ortonormalnej bazie $\phi_n(w)$, gdzie odpowiednio $w=x$ lub $w=y$ można udowodnić, że funkcja $g(x, y)$ daje się rozwinąć na szereg funkcji dwuwymiarowych $\Omega_{n,m}(x, y)$ będących iloczynami funkcji $\phi_n(x)$ i $\phi_m(y)$ „wszystkich przez wszystkie”:

$$\Omega_{n,m}(x, y)=\phi_n(x)\phi_m(y) \tag{1.20}$$

mianowicie:

$$g(x, y)=\sum_{n,m} Q_{n,m}\Omega_{n,m}(x, y) \tag{1.21}$$

gdzie liczby $Q_{n,m}$ są współczynnikami rozwinięcia, zaś $n, m=0,1,2,3 \ldots N-1$.

Dowód powyższego twierdzenia znajduje się w [1]. Współczynniki rozwinięcia $Q_{n,m}$ opisane mogą być równaniem:

$$Q_{n,m}=\int\int g(x, y)\phi_n(x)\phi_m(y)\,dydx \tag{1.22}$$

Baza $\Omega_{n,m}(x, y)$ jest bazą ortonormalną, ponieważ

$$\int\int [\Omega_{n,m}(x, y)]^2\,dxdy=1 \tag{1.23}$$

i gdy zachodzi choć jedna z nierówności $n\neq n'$ lub $m\neq m'$, to wówczas:

$$\int\int \Omega_{n,m}(x, y)\Omega_{n',m'}(x, y)\,dxdy=0 \tag{1.24}$$

Wyżej wymienione równanie (1.22) ukazuje zasadę obliczania współczynników rozwinięcia dla funkcji dwuwymiarowych i można przedstawić je w postaci:

$$Q_{n,m}=\int\int g(x, y)\Omega_{n,m}(x, y)\,dxdy \tag{1.25}$$

gdzie $\Omega_{n,m}(x, y)$ jest jedną z funkcji bazowych.

Jako aproksymację obrazu oryginalnego $f(x,y)$ na poziomie j rozumiemy obraz, którego przekroje są aproksymacjami na poziomie j - dają się rozwinąć w następujący sposób:

$$f^{(j)}(x,y_0)=\sum_{n=0}^{N(j)-1} r_n(y_0)\phi_{j,n}(x)$$

$$f^{(j)}(x_0,y)=\sum_{n=0}^{N(j)-1} r_n(x_0)\phi_{j,n}(y)$$

gdzie $\phi_{j,n}(w)$ są funkcjami skalującymi. Aproksymację dwuwymiarową oznacza się poprzez $f^{(j)}(x,y)$. Aproksymację można rozwinąć na szereg iloczynów jednowymiarowych funkcji skalujących:

$$f^{(j)}(x,y)=\sum_{n,m} f_{n,m}^{(j)}\phi_{j,n}(x)\phi_{j,m}(y) \tag{1.26}$$

gdzie liczby $f_{n,m}^{(j)}$ są współczynnikami rozwinięcia.

Baza przesuwna jest tworzona przez iloczyny funkcji skalujących $\phi_{j,n}(x)\phi_{j,m}(y)$ - $2^j n$ jest przesunięciem wzdłuż osi X, a $2^j m$ przesunięciem wzdłuż osi Y. Można je uważać za dwuwymiarowe funkcje skalujące. Bazy złożone w przypadku jednowymiarowym składają się w połowie z funkcji skalujących, a w połowie z falek. W przypadku dwuwymiarowym poprzez wymnożenie funkcji jednowymiarowych otrzymujemy cztery rodzaje funkcji bazowych:

$$\phi_{j+1,n}(x)\phi_{j+1,m}(y) \tag{1.27}$$

$$\phi_{j+1,n}(x)\psi_{j+1,m}(y) \tag{1.28}$$

$$\psi_{j+1,n}(x)\phi_{j+1,m}(y) \tag{1.29}$$

$$\psi_{j+1,n}(x)\psi_{j+1,m}(y) \tag{1.30}$$

Ponieważ pierwszy z powyższych rodzajów funkcji (funkcje skalujące) stanowi bazę przesuwną na poziomie $j+1$, a następne są dwuwymiarowymi falkami możemy uważać stworzoną przez nich bazę za bazę złożoną. Aproksymacja rozwinięta w tej bazie złożonej ma postać:

$$f^{(j)}(x,y)=f^{(j+1)}(x,y)+h^{(j+1)}(x,y)+v^{(j+1)}(x,y)+d^{(j+1)}(x,y) \tag{1.31}$$

gdzie:

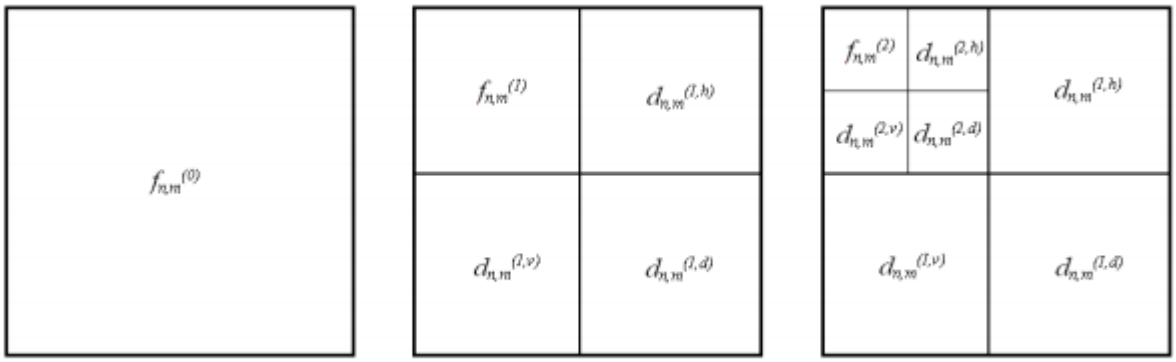
$$f^{(j+1)}(x,y)=\sum_{n,m} f_{n,m}^{(j+1)}\phi_{j+1,n}(x)\phi_{j+1,m}(y) \tag{1.32}$$

$$h^{(j+1)}(x,y)=\sum_{n,m} d_{n,m}^{(j+1,h)}\phi_{j+1,n}(x)\psi_{j+1,m}(y) \tag{1.33}$$

$$v^{(j+1)}(x,y)=\sum_{n,m} d_{n,m}^{(j+1,v)}\psi_{j+1,n}(x)\phi_{j+1,m}(y) \tag{1.34}$$

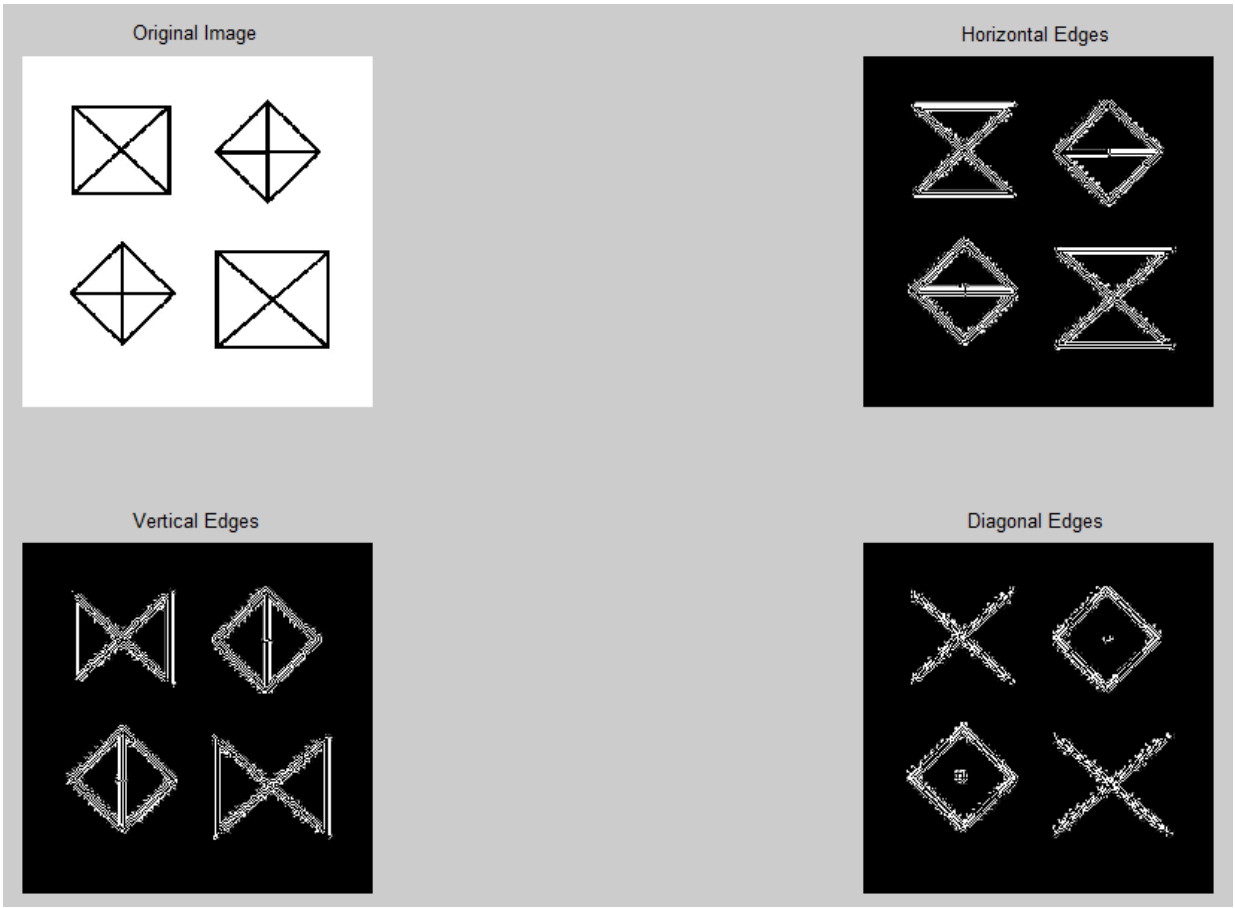
$$d^{(j+1)}(x,y)=\sum_{n,m} d_{n,m}^{(j+1,d)}\psi_{j+1,n}(x)\psi_{j+1,m}(y) \tag{1.35}$$

gdzie $f_{n,m}^{(j+1)}, d_{n,m}^{(j+1,h)}, d_{n,m}^{(j+1,v)}, d_{n,m}^{(j+1,d)}$ to współczynniki rozwinięcia oraz . $n,m=0,1,2,3\ldots\frac{1}{4}N(j)-1$.



Rysunek 5: Rozmieszczenie współczynników rozwinięcia f. dwóch zmiennych po pierwszej i drugiej iteracji

Jak można zauważyć skutkiem dekompozycji na poziomie j jest zbiór 4 komponentów, które można interpretować jako sub-obrazy o dwukrotnie mniejszej rozdzielczości w stosunku do obrazu dekomponowanego – są to – aproksymacja na poziomie o jeden wyższym oraz trzy zawierających detale. Obrazy te nie są identyczne – obraz $h^{(j+1)}(x, y)$ pokazuje krawędzie poziome nie pokazując pionowych, obraz $d^{(j+1)}(x, y)$ pokazuje krawędzie pionowe nie pokazując poziomych, natomiast obraz $v^{(j+1)}(x, y)$ ukazuje krawędzie skośne.

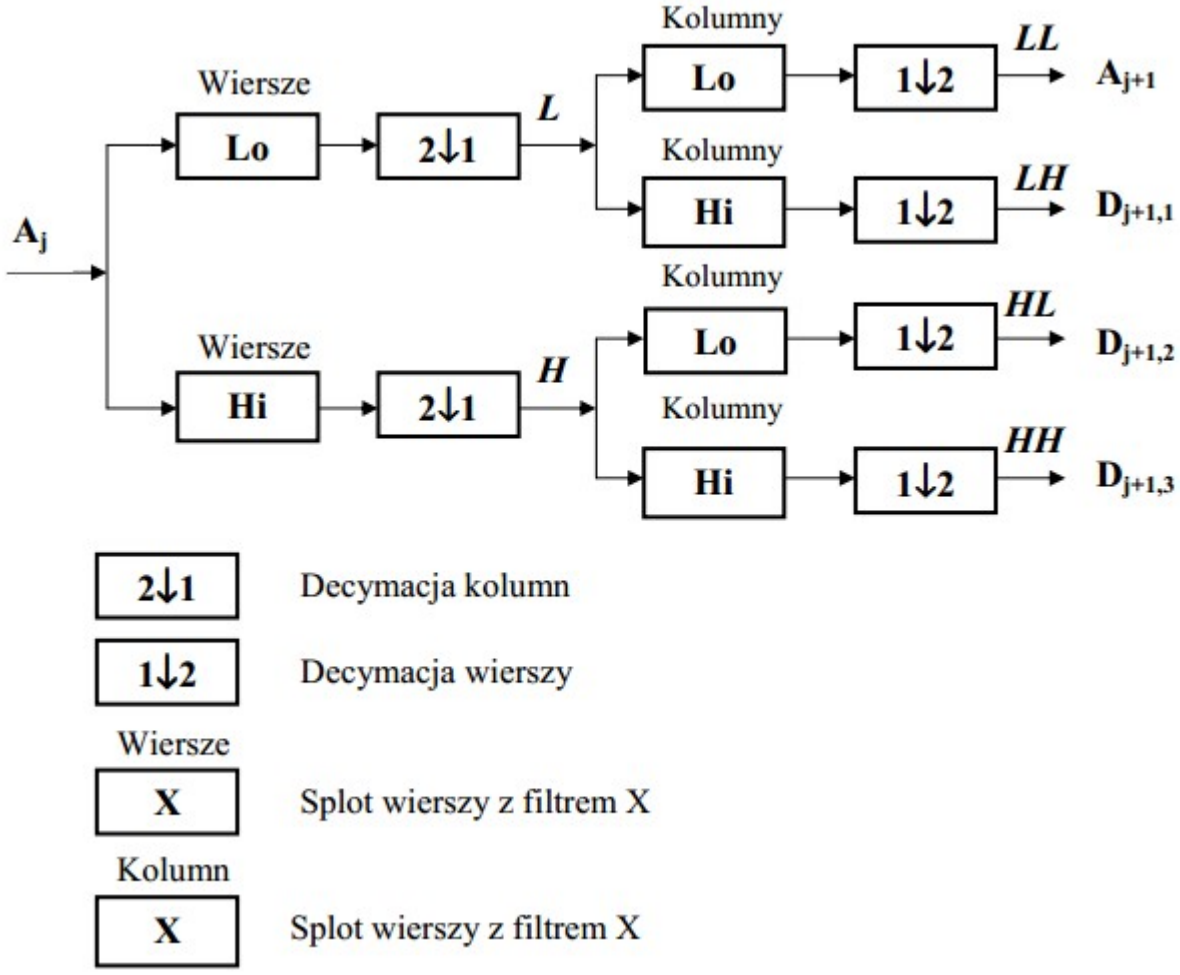


Rysunek 6: Składowe aproksymacji zawierające detale

Obraz składowych aproksymacji zawierających detale osiągnięto wykorzystując falkę db4. Powyższy obraz otrzymano za pomocą programu aproksymacja.m analizując obraz test.jpg – zarówno obraz jak i program są dołączone do projektu.

Dwuwymiarowa dyskretna transformata falkowa

Podobnie jak w przypadku jednowymiarowym do analizy sygnału (obrazu) można zastosować dyskretną transformatę falkową. Analiza macierzy dwuwymiarowej, jaką jest np. monochromatyczny obraz, dokonywana jest najpierw na wierszach, a następnie na kolumnach. W ten sposób w pojedynczym etapie dekompozycji obraz jest rozkładany przez filtry dolno- i górnoprzepustowe na cztery podobrazy – jedną aproksymację oraz trzy detale. Podobraz LL jest pomniejszoną kopią oryginalnego obrazu. Natomiast pozostałe (LH, HL oraz HH) wydobywają zmieniające się składniki obrazu, odpowiednio w kierunku poziomym, pionowym oraz diagonalnym.



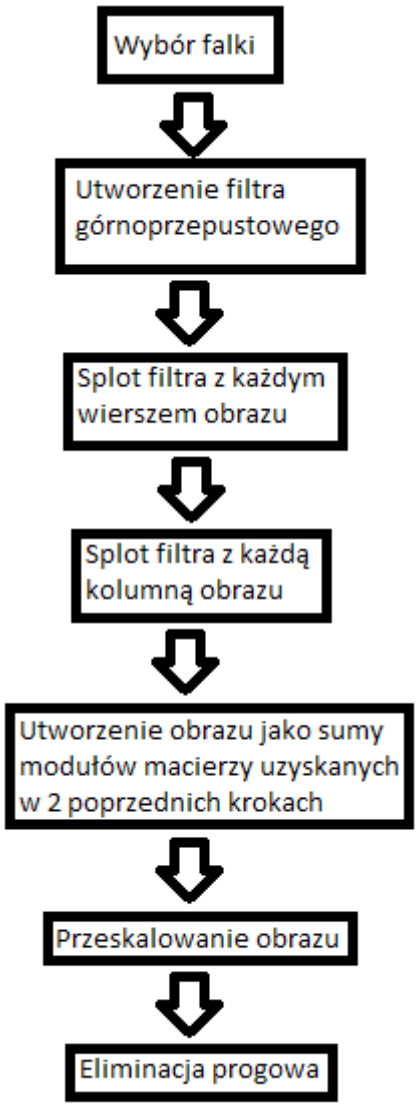
Rysunek 7: Schemat jednopoziomowej dekompozycji sygnału 2D

Tak jak w przypadku DWT jednowymiarowej, tak i przy przetwarzaniu sygnałów dwuwymiarowych, proces ten może być powtórzony przez zastosowanie rekurencji. Można w ten sposób uzyskać bardziej szczegółowe dane dotyczące przetwarzanego obrazu.

2. Teoretyczny opis metod

2.1 Row-Column Method

Jest to zaproponowana przez dr. Yahia Al-Halabi oraz Hesham Jondi Abd na łamach Journal of Theoretical and Applied Information Technology [7] metoda, która opiera się na następującym algorytmie:



Rysunek 8: Uproszczony schemat algorytmu Row-Column

Krok 1. Wybór rodzaju oraz rzędu falki

Krok 2. Stworzenie filtru górnoprzepustowego (high-pass decomposition filter) charakterystycznego dla wybranej falki

Krok 3. Dla każdego wiesza obrazu dokonać splotu filtra z danym wierszem:

RDiff = Convolution (HI_D, IM(i, :));

Krok 4. Dla każdej kolumny obrazu dokonać splotu filtra z daną kolumną:

CDiff = Convolution (HI_D, IM(:, j));

RDiff i CDiff – dwuwymiarowe macierze przedstawiające różnice między pikselami odpowiednio w wierszach i kolumnach obrazu

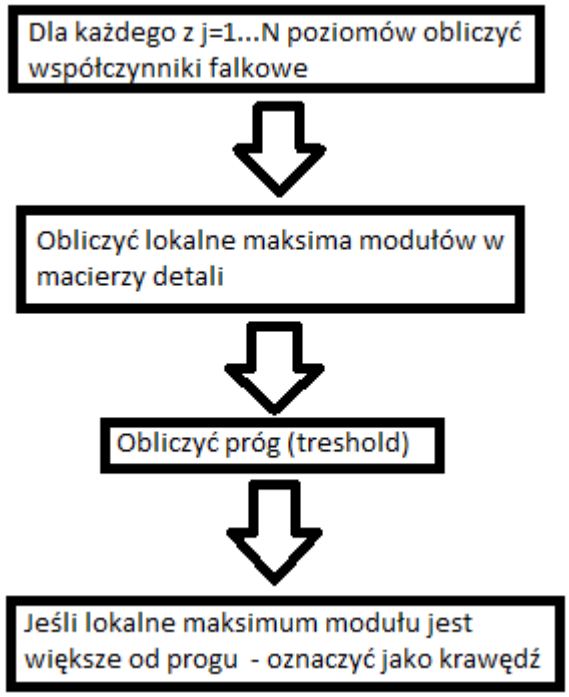
Krok 5. Utworzyć obraz jako sumę wartości bezwzględnych macierzy RDiff i CDiff:

EM = |RDiff (i, j))| + |CDiff (i, j))|

Krok 6. Przeskalować obraz tak, by wartości w macierzy były z przedziału <0,1>

Krok 7. Zastosować eliminację progową (tresholding) w celu wydobycia najważniejszych pikseli reprezentujących krawędzie.

2.2 Wieloskalowa detekcja krawędzi (multiscale edge detection) z wykorzystaniem uproszczonego algorytmu „à trous”



Rysunek 9: Wieloskalowa detekcja krawędzi

Wynikiem transformaty falkowej są 2 obrazy – aproksymacja oraz detale. W macierzy zawierającej współczynniki detali krawędzie opisywane są przez wartości odległe od zera, zarówno dodatnie jak i ujemne. Dlatego też można skorzystać z wartości bezwzględnej z tej macierzy.

Podczas selekcjonowania najbardziej istotnych pikseli obliczamy lokalne maksima – czyli piksele, których wartość jest większa od najbliższego otoczenia oraz porównujemy je z progiem – czyli średnią wartością piksela w całej macierzy.

Krok 1. Obliczenie współczynników falkowych

```
[~, detail] = a_trous(I, N);
```

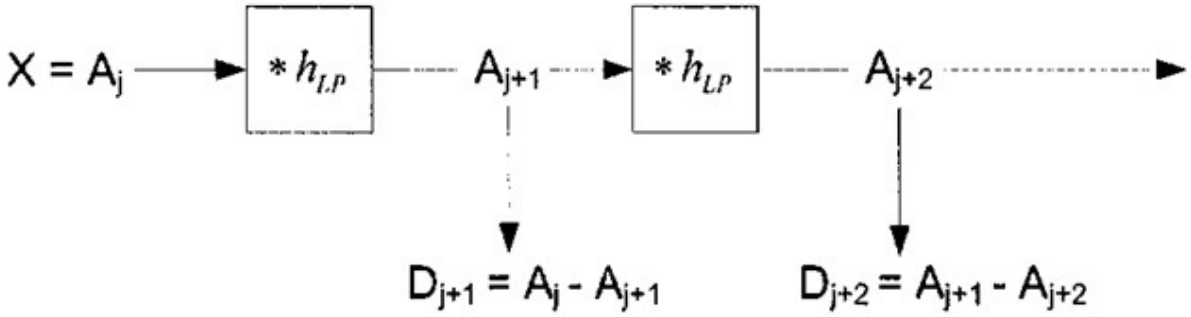
Krok 2. Obliczenie modułu z falkowych współczynników detali

```
D = abs( detail(:,:,N) );
```

Krok 3. Oznaczenie pikseli o wartości większej od najbliższego otoczenia oraz większej od średniej całego obrazu (dynamiczne obliczanie progu) jako krawędź

```
J = (D>filter2(ones(3)/9,D)).*(D>mean2(D));
```

Algorytm „à trous” polega na uzyskiwaniu współczynników detali jako różnic pomiędzy kolejnymi aproksymacjami. Kolejne aproksymacje uzyskuje się w wyniku splotu filtra z aproksymacją niższego rzędu:



Rysunek 10: Schemat algorytmu "à trous"

Algorytm „a trous”

Krok 1. Utworzenie filtru dolnoprzepustowego

```
B3 = [1/16 1/4 3/8 1/4 1/16];
h = B3'*B3;
```

Krok 2. Splot filtru z aproksymacją niższego rzędu

```
approx(:,:,level) = conv2(A, h, 'same');
```

2.3 Metody korzystające z modyfikacji współczynników aproksymacji

J. Petrova, E. Hostalkova z Institute of Chemical Technology w Pradze w pracy "Edge detection in medical images using the wavelet transform" [8] zaproponowały 2 metody opierające się na modyfikacji współczynników aproksymacji. Zazwyczaj po przeprowadzeniu dyskretnej transformaty falkowej w dalszej analizie korzysta się z otrzymanych obrazów zawierających detale, nie korzystając z otrzymanej aproksymacji. Autorki powyższej pracy zaproponowały 2 metody modyfikacji współczynników aproksymacji:

1) zastąpienie wszystkich współczynników aproksymacji zerami
2) modyfikacja współczynników aproksymacji z wykorzystaniem prostych detektorów krawędzi

Zaimplementowano drugą metodę, w której po przeprowadzeniu DWT otrzymany obraz aproksymacji poddano działaniu filtra wykrywającego krawędzie. Transformata falkowa została przeprowadzona algorytmem „à trous".

Rysunek 11: Metoda korzystająca z modyfikacji współczynników aproksymacji

Krok 1. Dyskretna transformata falkowa

```
[approx, ~] = a_trous(I, N);
```

Krok 2. Detekcja krawędzi algorytmem Canny

```
J=edge(approx,'canny');
```

2.4 Wieloskalowa detekcja krawędzi (multiscale edge detection)

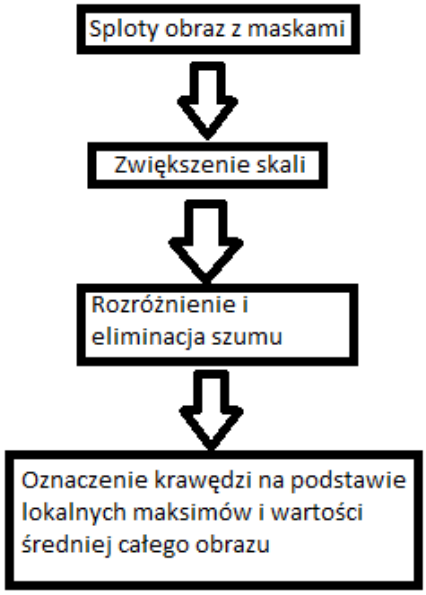
Jun Li w „A Wavelet Approach to Edge Detection” [6] zaproponował wieloskalową detekcję krawędzi, w której podobnie jak w opisanej w punkcie 2.2 końcowa selekcja pikseli krawędzi polega na porównaniu wartości każdego piksela z jego najbliższym otoczeniem i średnią wartością piksela całego obrazu. Główną różnicą między metodą Jun Li, a zaproponowaną w punkcie 2.2 jest sposób liczenia współczynników detali – zamiast transformaty falkowej w posiatci algorytmu „à trous” Jun Li zastosował odpowiednie dwuwymiarowe dyskretne sploty z macierzy 3x3, będącymi tzw. maskami Sobela z obrazem. Dwie z tych macierzy do maski Sobela dla 180° oraz 270°. Trzecia to jedna z macierzy wchodzącej w skład trójwymiarowego operatora Sobela [10]. Przed ostateczną selekcją krawędzi pikseli konieczne było również rozróżnienie i eliminacja szumu.

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

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

1	2	1
2	4	2
1	2	1

Rysunek 12: Maski Sobela wykorzystywane w metodzie autorstwa Jun Li



Rysunek 13: Wieloskalowa detekcja krawędzi zaproponowana przez Jun Li

Krok 1. Utworzenie masek

```
h1=[1 0 -1; 2 0 -2; 1 0 -1];
h2=[1 2 1;0 0 0;-1 -2 -1];
v=[1 2 1;2 4 2; 1 2 1];
```

Krok 2. Splot obrazu z maskami

```
a1=filter2(h1,a);
a2=filter2(h2,a);
```

Krok 3. Zwiększenie skali

```
ah1=filter2(v,a1)*p;
ah2=filter2(v,a2)*p;
ns=nsh;
```

Krok 4. Rozróżnienie i eliminacja szumu

```
a1=a1.*(abs(a1)<=abs(ah1))+ah1.*(abs(a1)>abs(ah1));
a2=a2.*(abs(a2)<=abs(ah2))+ah2.*(abs(a2)>abs(ah2));
```

Krok 5. Obraz końcowy jako suma obrazów a1 i a2

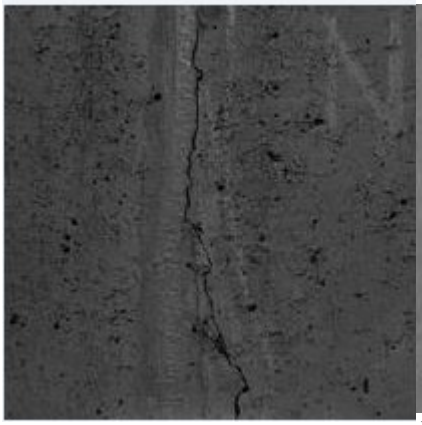
```
e=(a1.^2+a2.^2);
```

Krok 6. Oznaczenie krawędzi, eliminacja progowa

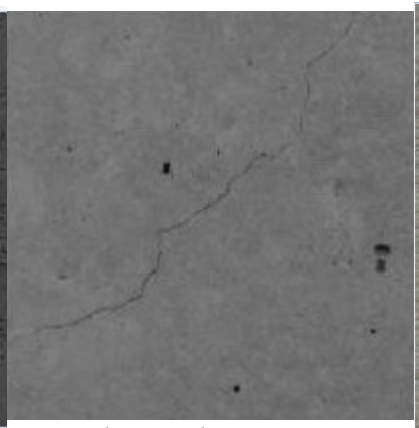
```
y=(e>(filter2((ones(10))/100,e))).*(e>mean2(e));
```

3. Testy poszczególnych metod

Testy przeprowadzono na następujących obrazach:



Rysunek 15: Obraz1.jpg o wymiarach 206x206 pikseli

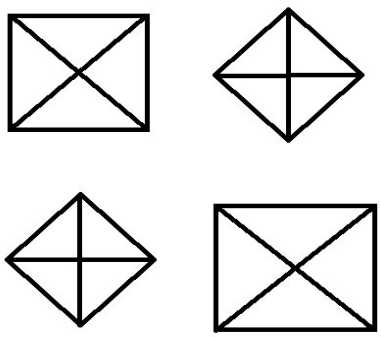


Rysunek 16: Obraz2.jpg o wymiarach 206x206 pikseli

Rysunek 14: Obraz3.jpg o wymiarach 768x1024 piksele

oraz 2 nieprzedstawiających pęknięcia:

Rysunek 17: Lena.jpg o wymiarach 512x512 pikseli



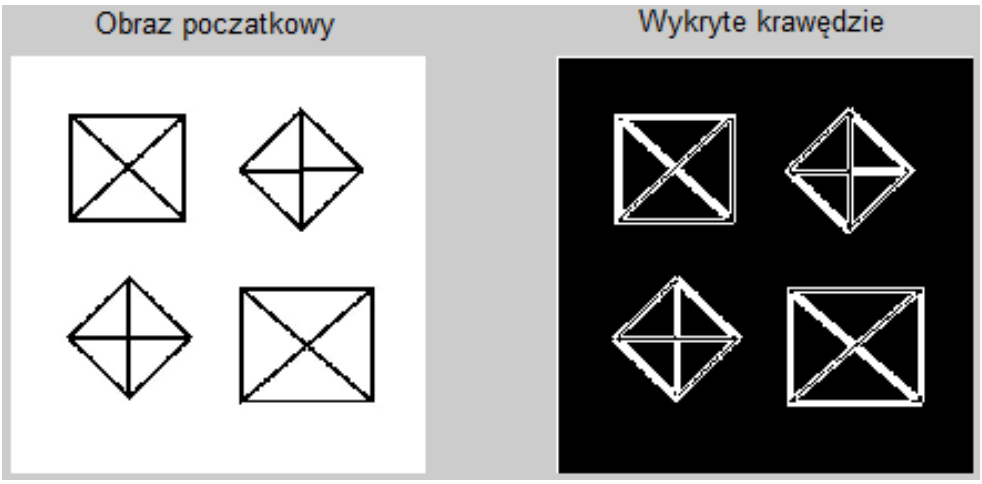
Rysunek 18: Test.jpg o wymiarach 512x512 pikseli

Wszystkie obrazy przed analizą są konwertowane do obrazu w skali szarości.

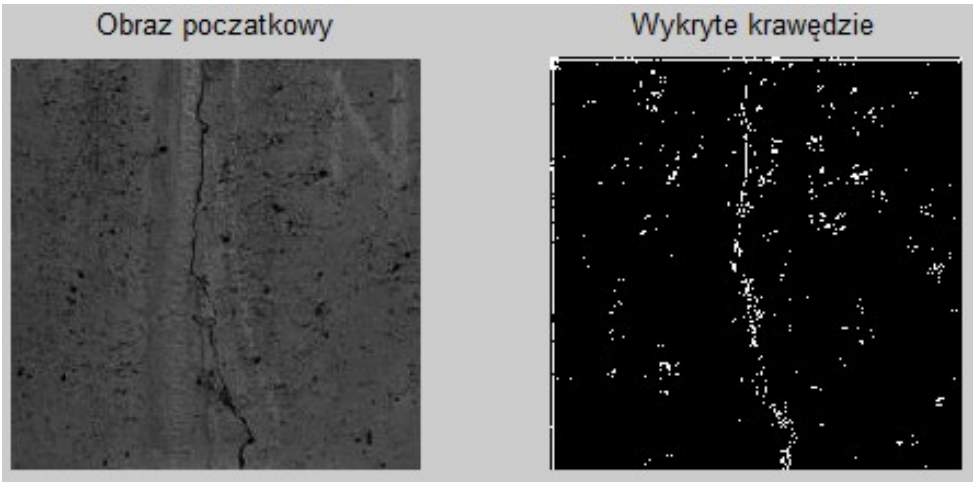
3.1 Row-Column Method

- Falka db4

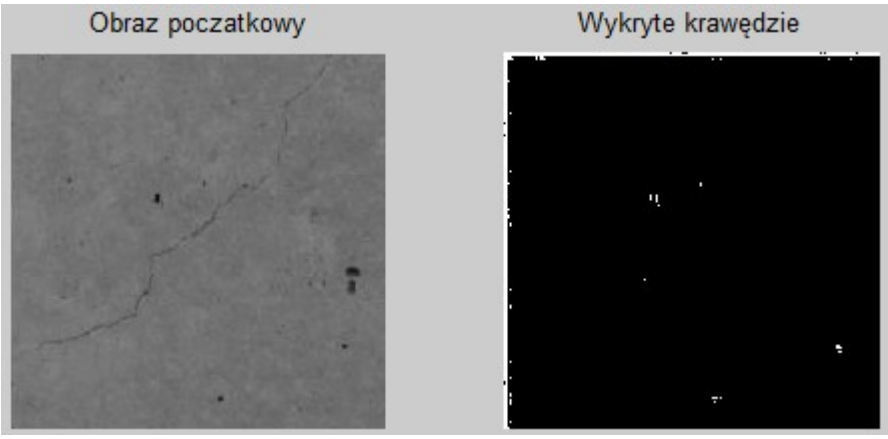
Rysunek 19: Obraz lena.jpg Falka: db4



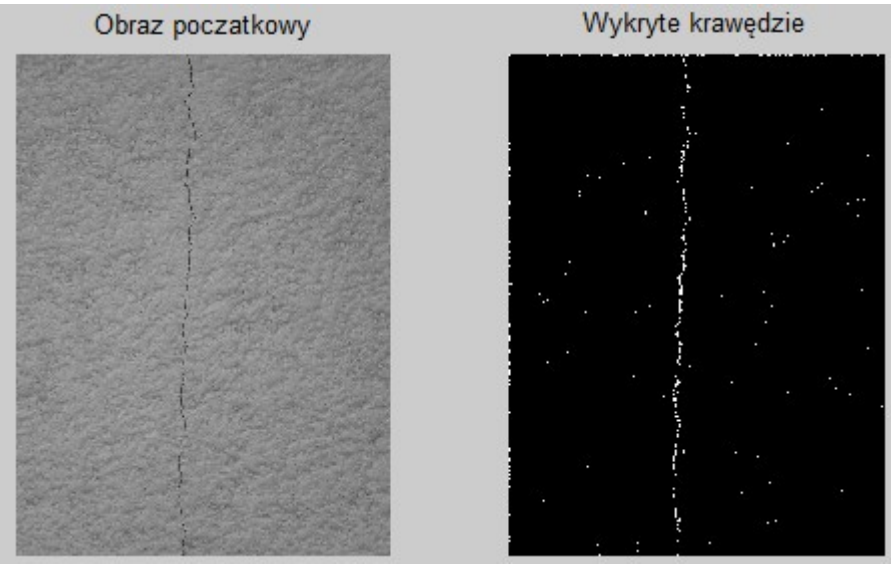
Rysunek 20: Obraz: test.jpg Falka: db4



Rysunek 21: Obraz: obraz1.jpg Falka: db4

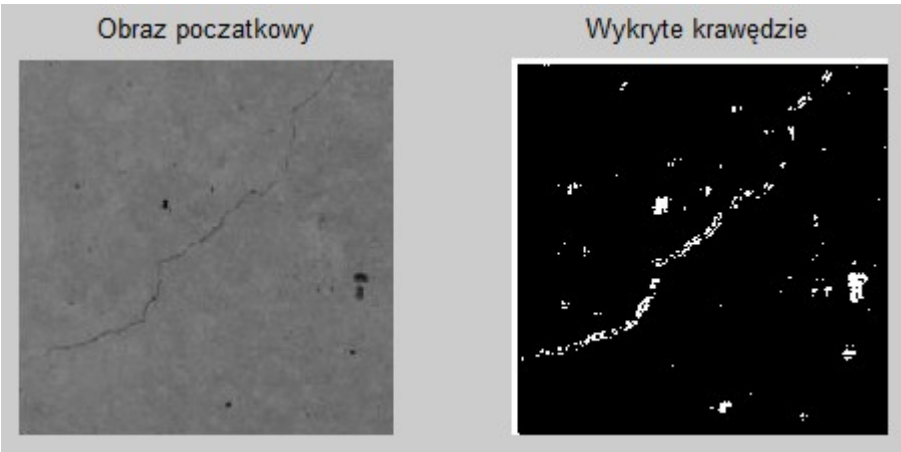


Rysunek 22: Obraz: obraz2.jpg Falka: db4



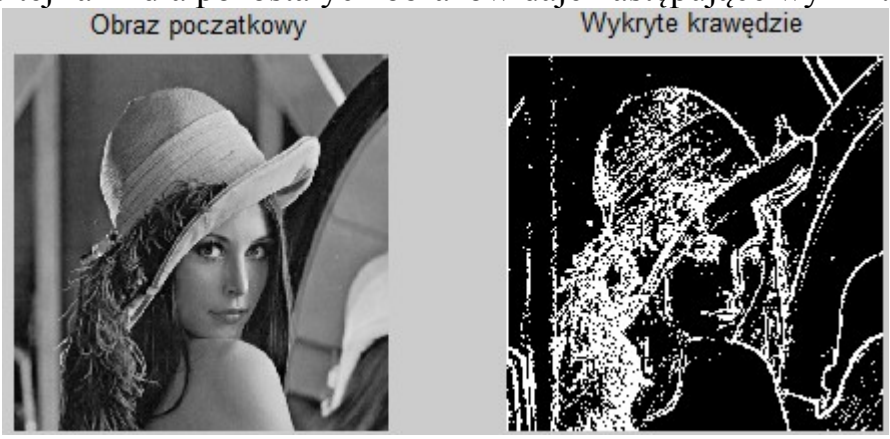
Rysunek 23: Obraz: obraz3.jpg Falka: db4

Można zauważyć, że w przypadku obrazu, gdzie krawędź nie wyróżnia się bardzo na tle algorytm przy zastosowanej falce db4 nie poradził sobie z wykryciem krawędzi. Można jednak zmienić falkę na rbio3.1 aby osiągnąć zadowalające rezultaty.

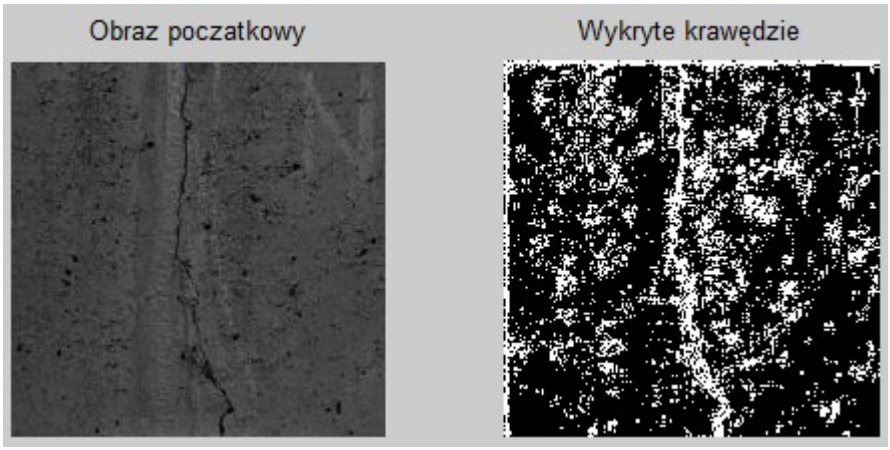


Rysunek 24: Obraz: obraz2.jpg Falka rbio3.1

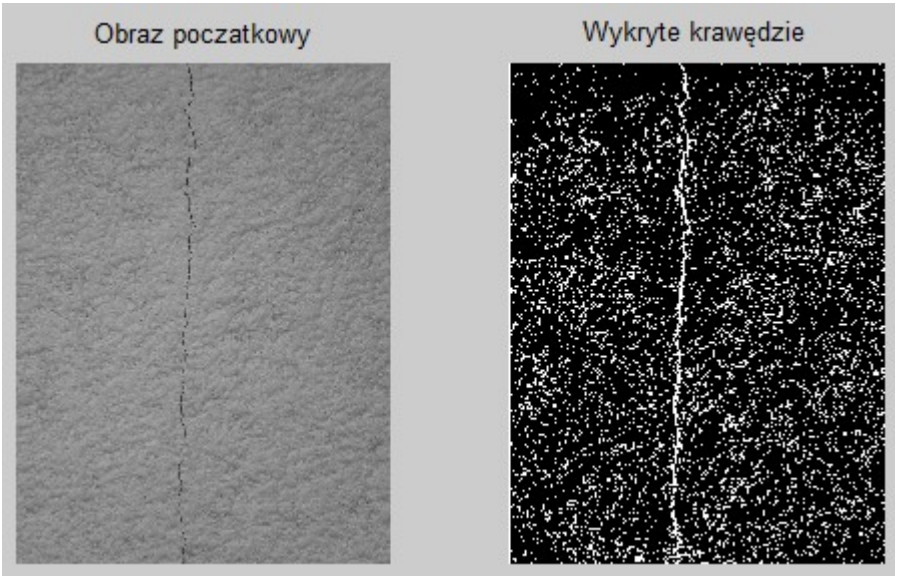
Zastosowanie tej falki dla pozostałych obrazów daje następujące wyniki:



Rysunek 25: Obraz: lena.jpg Falka: rbio3.1



Rysunek 26: Obraz: obraz1.jpg Falka: rbio3.1

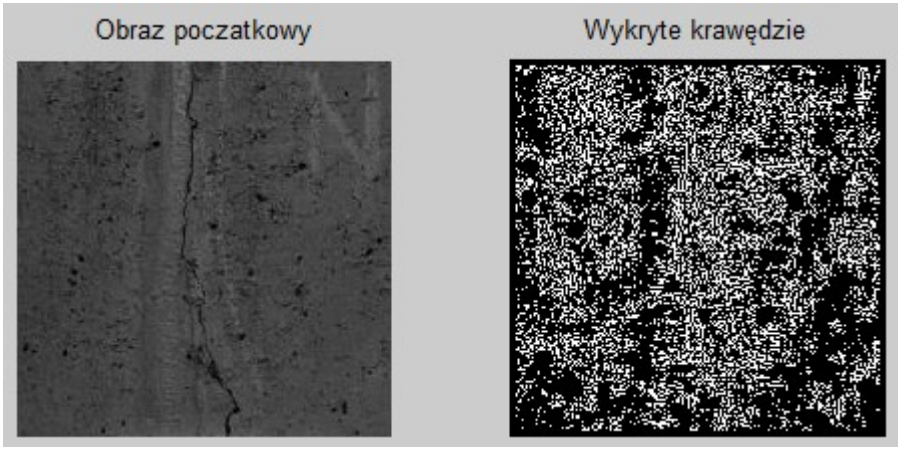


Rysunek 27: Obraz: obraz3.jpg Falka:rbio3.1

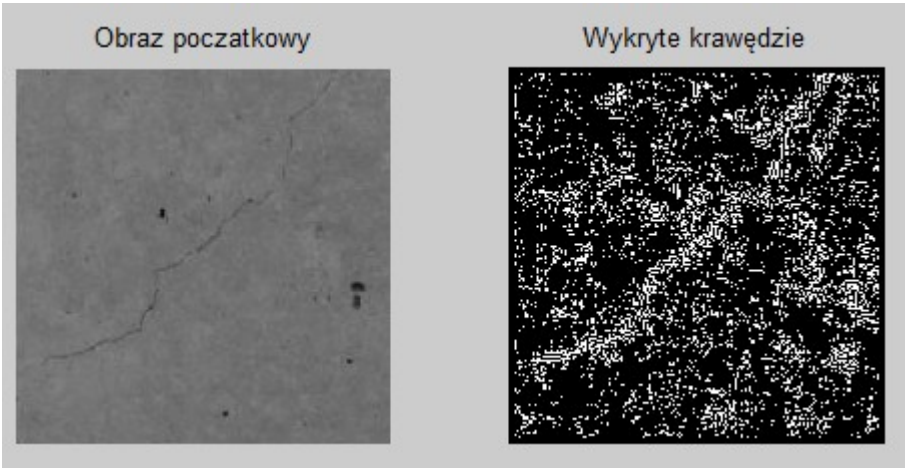
Jak widać zastosowanie tej falki nie w każdym przypadku daje lepsze wyniki – w przypadku obrazów obraz1 i obraz3 wykrywa również sporo szumów. Dlatego też w każdym przypadku należy sprawdzić ten algorytm dla kilku różnych falek i wybrać tą, która w najlepszy sposób wykryje krawędzie.

3.2 Wieloskalowa detekcja krawędzi (multiscale edge detection) z wykorzystaniem uproszczonego algorytmu „à trous”

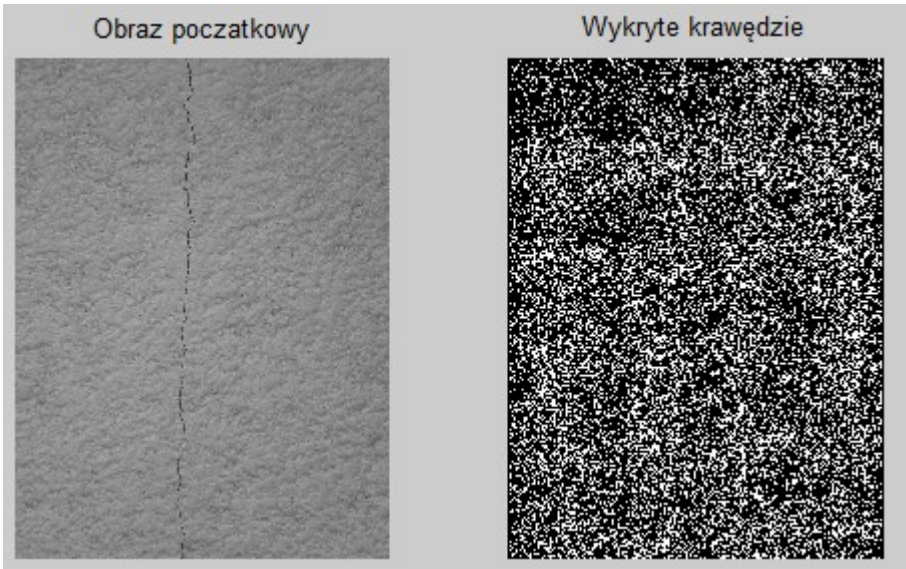
Rysunek 28: Obraz: lena.jpg Rząd dekompozycji: 1



Rysunek 29: Obraz: obraz1.jpg Rząd dekompozycji: 1



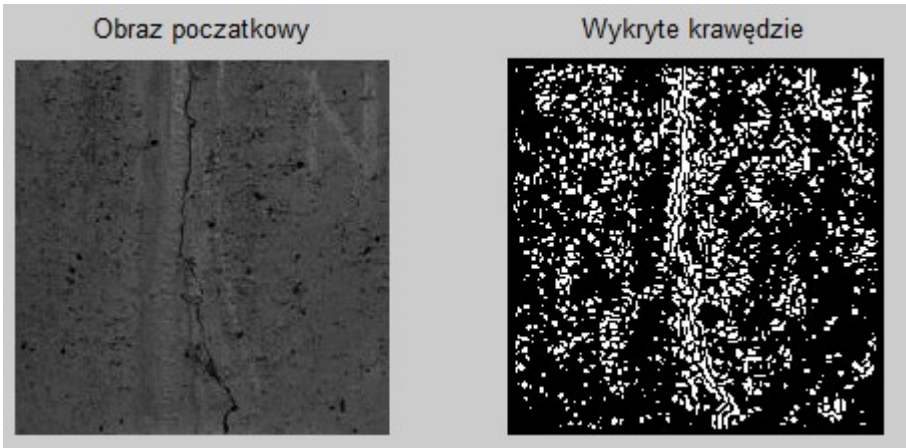
Rysunek 30: Obraz: obraz2.jpg Rząd dekompozycji: 2



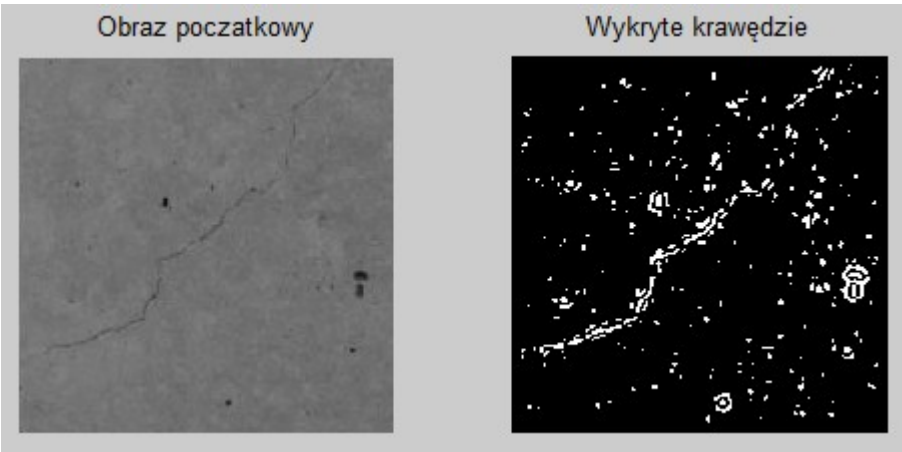
Rysunek 31: Obraz: obraz3.jpg Rząd dekompozycji: 1

Przy rzędzie dekompozycji równym 1 algorytm wykrywa bardzo dużo szumów, leczy wraz z zwiększaniem rzędu dekompozycji otrzymujemy tylko najbardziej istotne krawędzie.

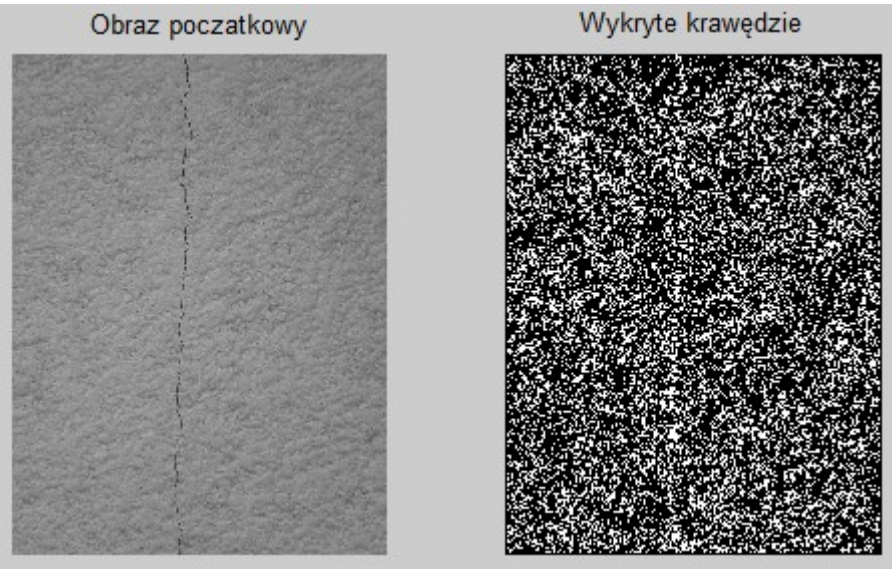
Rysunek 32: Obraz: lena.jpg Rząd dekompozycji: 3



Rysunek 33: Obraz: obraz1.jpg Rząd dekompozycji: 3

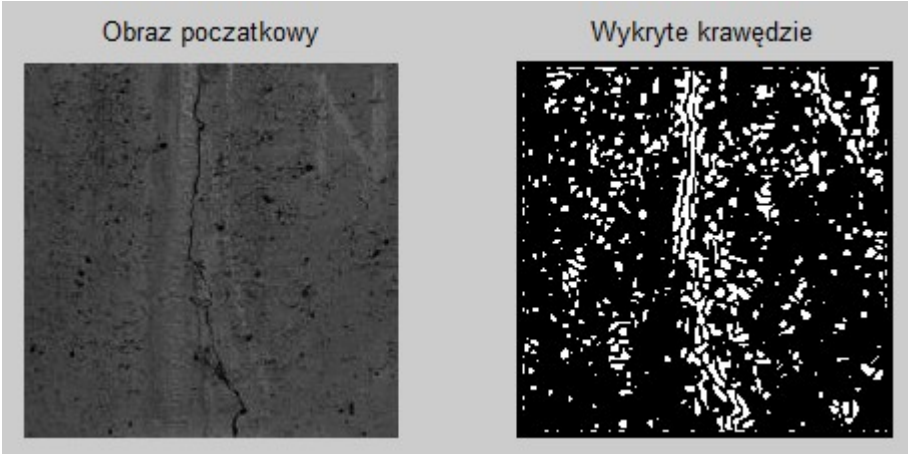


Rysunek 34: Obraz: obraz2.jpg Rząd dekompozycji: 3

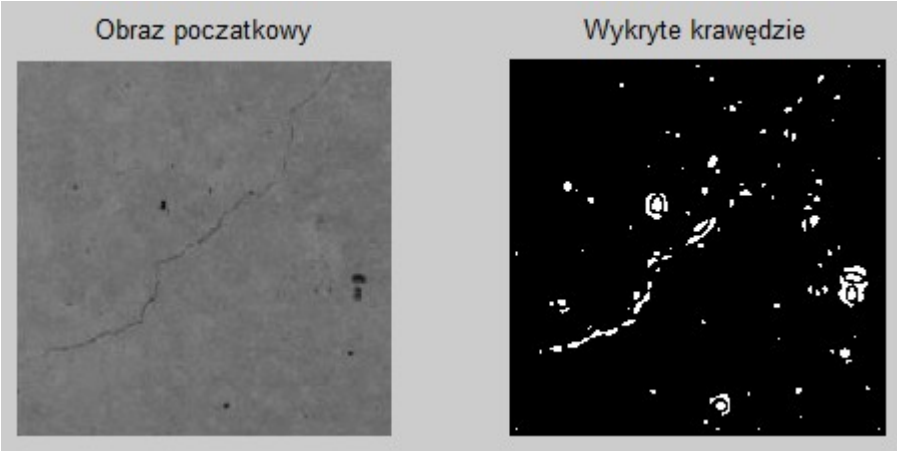


Rysunek 35: Obraz: obraz3.jpg Rząd dekompozycji: 3

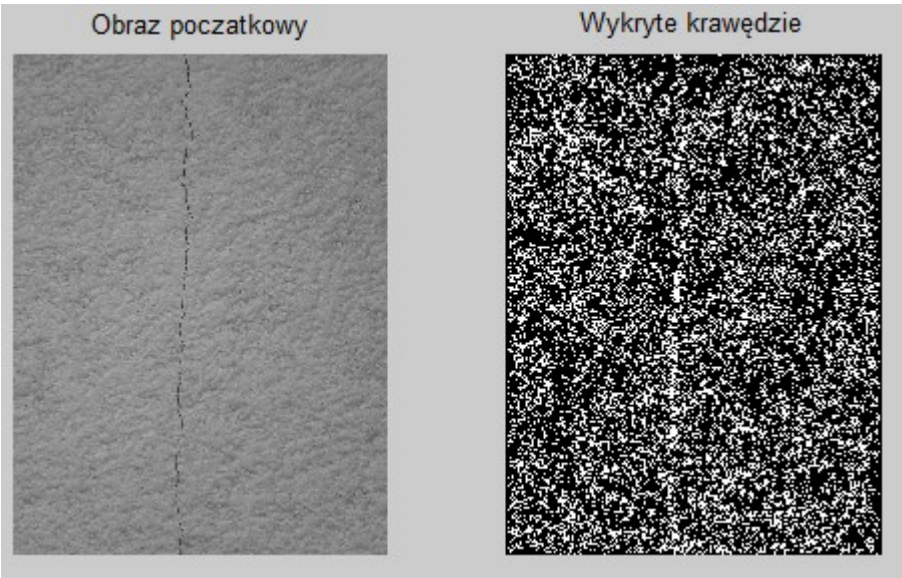
Rysunek 36: Obraz: lena.jpg Rząd dekompozycji: 5



Rysunek 37: Obraz: obraz1.jpg Rząd dekompozycji: 5

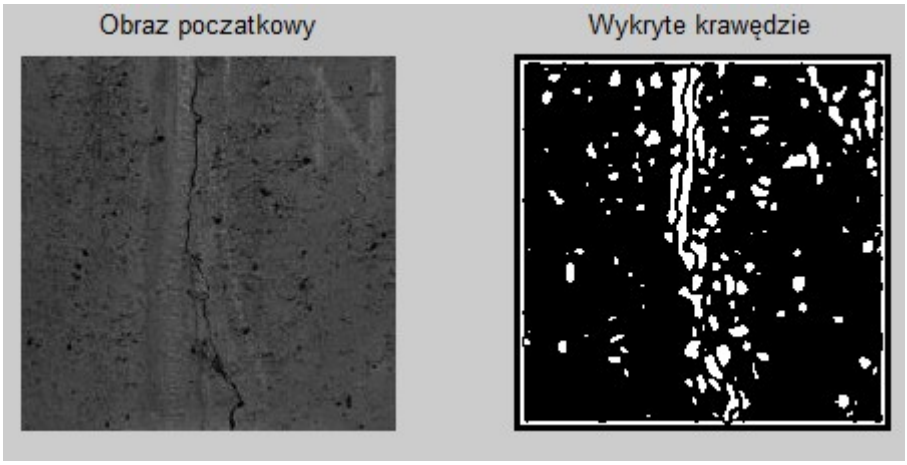


Rysunek 38: Obraz: obraz2.jpg Rząd dekompozycji: 5

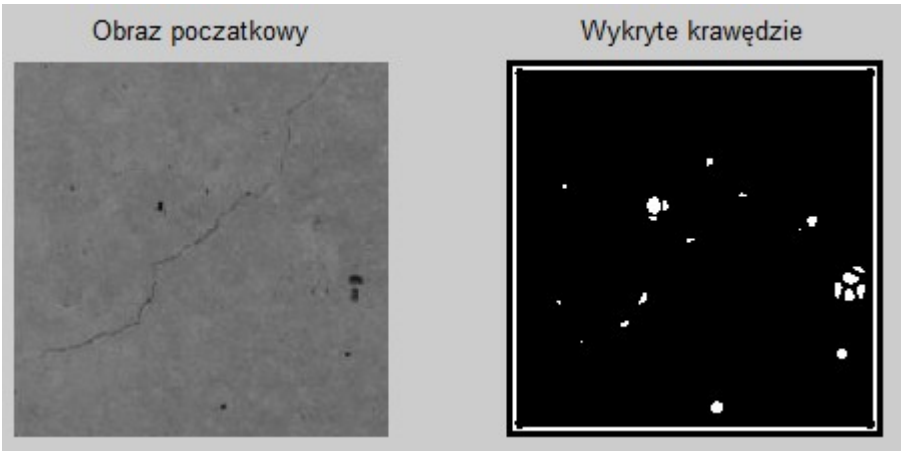


Rysunek 39: Obraz: obraz3.jpg Rząd dekompozycji: 5

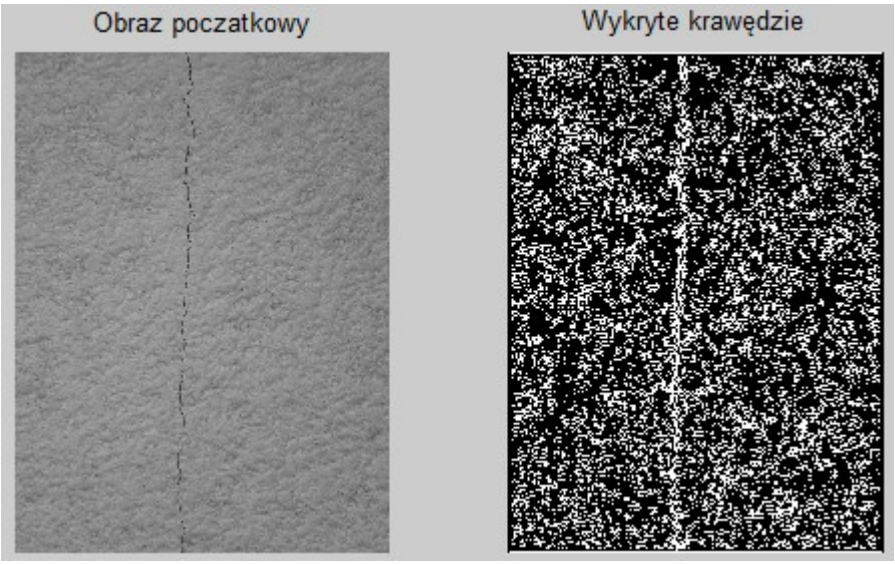
Rysunek 40: Obraz: lena.jpg Rząd dekompozycji: 10



Rysunek 41: Obraz: obraz1.jpg Rząd dekompozycji: 10

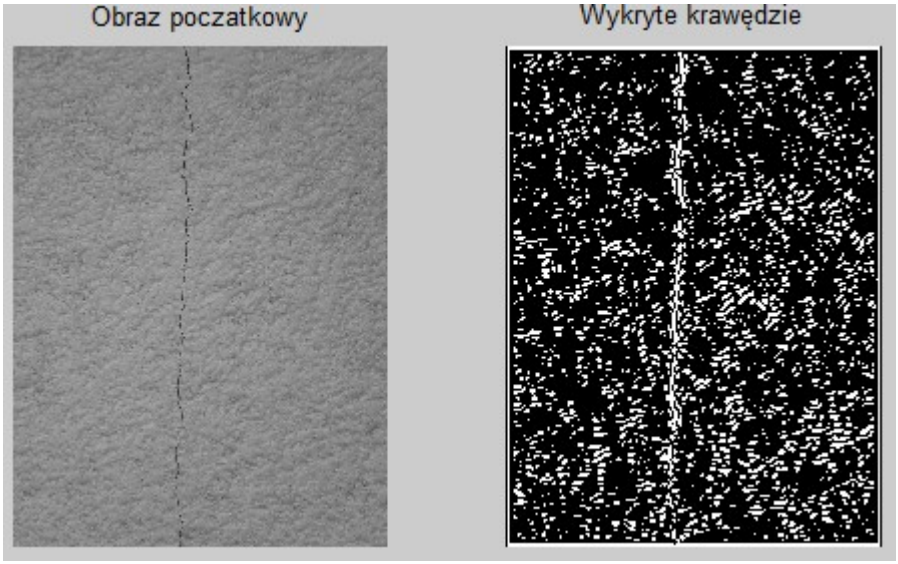


Rysunek 42: Obraz: obraz2.jpg Rząd dekompozycji: 10



Rysunek 43: Obraz: obraz3.jpg Rząd dekompozycji: 10

W przypadku obraz2.jpg zwiększenie rzędu dekompozycji może spowodować, że algorytm nie wykryje żadnych krawędzi. Z kolei w przypadku obrazu mocno zaszumionego jakim jest obraz3.jpg nawet bardzo duże zwiększenie rzędu dekompozycji nie eliminuje szumów:

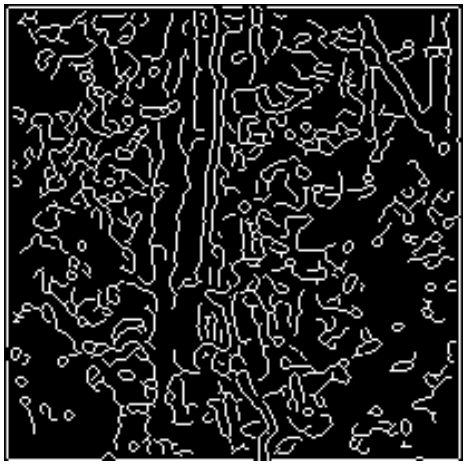


Rysunek 44: Obraz: obraz3.jpg Rząd dekompozycji: 30

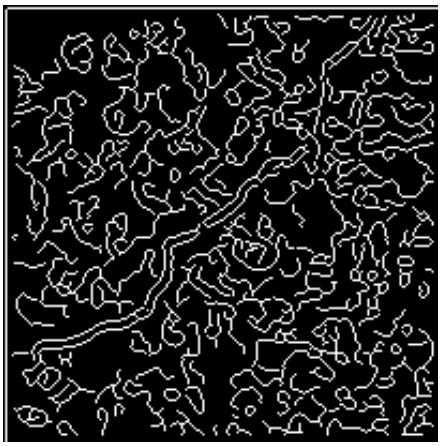
:

3.3 Metoda korzystająca z modyfikacji współczynników aproksymacji

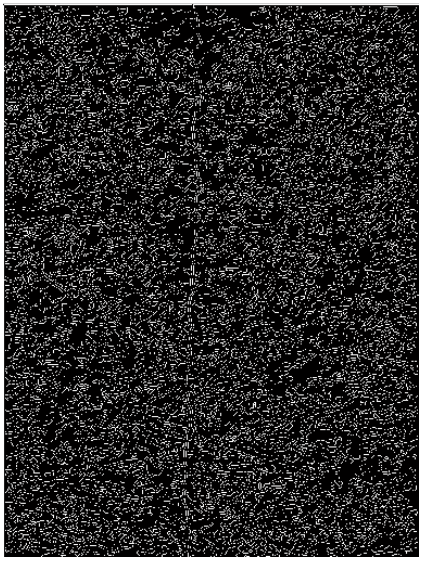
Rysunek 45: Obraz lena.jpg, rząd dekompozycji: 1



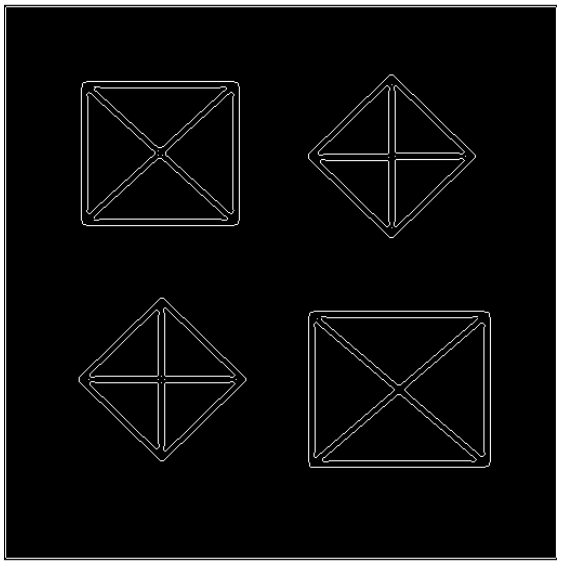
Rysunek 46: Obraz: obraz1.jpg Rząd dekompozycji: 1



Rysunek 48: Obraz: obraz2.jpg Rząd dekompozycji: 1

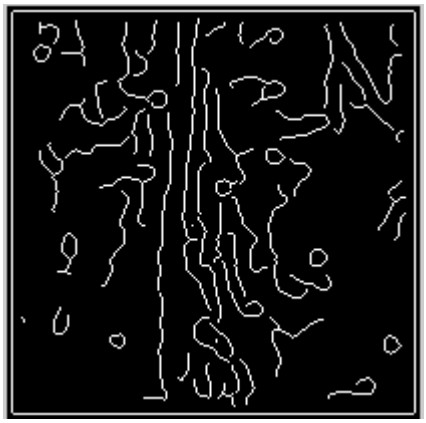


Rysunek 47: Obraz: obraz3.jpg Rząd dekompozycji: 1



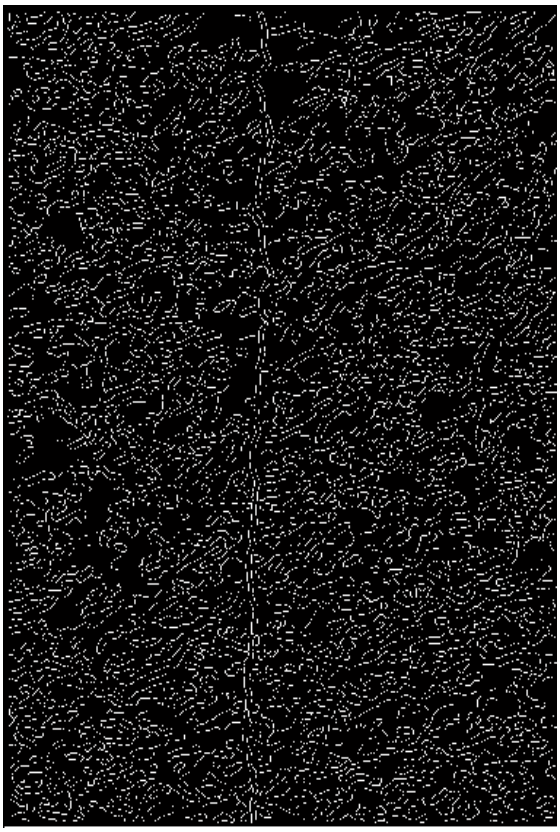
Rysunek 49: Obraz: test.jpg Rząd dekompozycji: 1

Algorytm bardzo dobrze znajduje krawędzie na obrazach niezaszumionych, lecz z zaszumionymi nie radzi sobie najlepiej. Zwiększenie rzędu dekompozycji nie zawsze poprawia w sposób zadowalający osiągnięte rezultaty:



Rysunek 50: Obraz: obraz1.jpg Rząd dekompozycji: 8

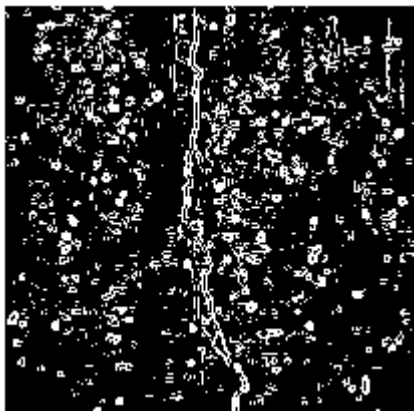
Rysunek 51: Obraz: obraz2.jpg Rząd dekompozycji: 8



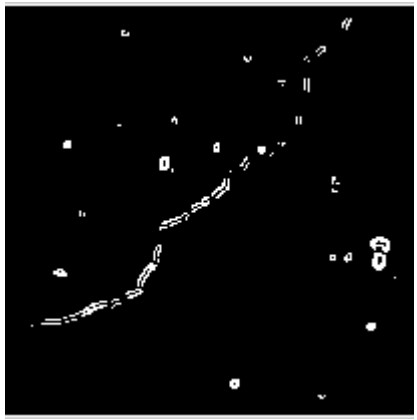
Rysunek 52: Obraz: obraz3.jpg Rząd dekompozycji: 8

3.4 Wieloskalowa detekcja krawędzi (multiscale edge detection)

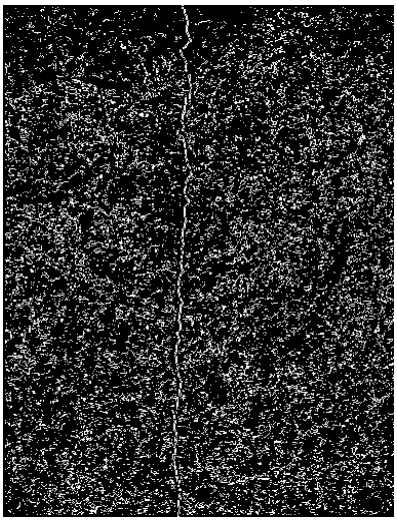
Rysunek 53: Obraz: lena.jpg Skala: 1



Rysunek 54: Obraz: obraz1.jpg Skala:1

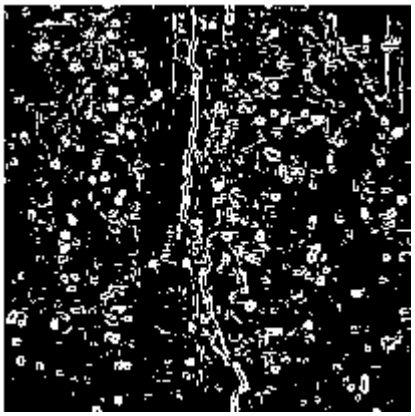


Rysunek 56: Obraz: obraz2.jpg Skala:1

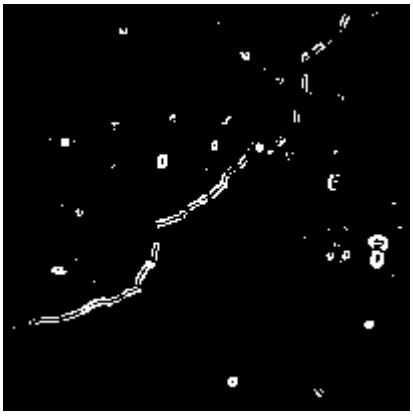


Rysunek 55: Obraz: obraz3.jpg Skala:1

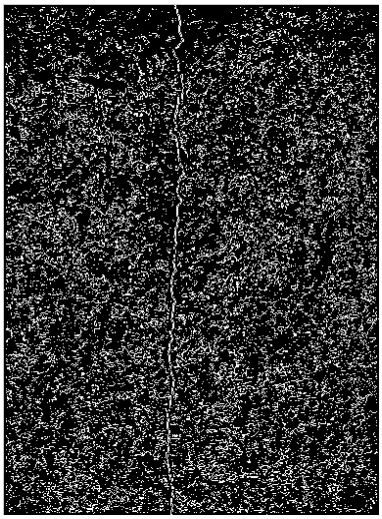
Rysunek 57: Obraz: lena.jpg Skala:3



Rysunek 58: Obraz: obraz1.jpg Skala:3



Rysunek 59: Obraz: obraz2.jpg Skala:3



Rysunek 60: Obraz: obraz3.jpg Skala:3

Zwiększanie skali powoduje zwiększenie ilości szumu na obrazie.

4.Wnioski

Subiektywna ocena przeprowadzonych testów

Zebrano i oceniono wyniki testów. Obraz kontrolny 1 to standardowy obraz często używany w przetwarzaniu obrazów – lena.jpg. Obraz kontrolny 2 to obraz1.jpg – przedstawiający pęknięcie, odróżniające się na tle oraz stosunkowo mało zaszumiony. Obraz o słabym kontraście to obraz2.jpg, gdzie odcień pęknięcia oraz tła nie różnią się znacząco od siebie. Obraz zaszumiony to obraz3.jpg, gdzie faktura powierzchni powoduje kłopoty

Metoda	Obraz kontrolny 1	Obraz kontrolny 2	Obraz o słabym kontraście	Obraz zaszumiony
Row-Column Method falka db4	zły	przeciętny	bardzo zły	dobry
Row-Column Method falka rbio3.1	przeciętny	zły	dobry	przeciętny
Algorytm „a trous” Rząd dekompozycji: 1	zły	bardzo zły	bardzo zły	bardzo zły
Algorytm „a trous” Rząd dekompozycji: 3	przeciętny	zły	dobry	bardzo zły
Algorytm „a trous” Rząd dekompozycji: 5	przeciętny	przeciętny	przeciętny	bardzo zły
Algorytm „a trous” Rząd dekompozycji:10	dobry	dobry	bardzo zły	zły
Metoda korzystająca z modyfikacji współ. aproksymacji	dobry	zły	zły	bardzo zły
Wieloskalowa detekcja Skala: 1	bardzo dobry	zły	dobry	zły
Wieloskalowa detekcja Skala: 3	bardzo dobry	zły	dobry	zły

Wnioski otrzymane z otrzymanych wyników zebrano poniżej:

1. Zaimplementowane algorytmy pozwalają na detekcję krawędzi. Nie są jednak w pełni skuteczne – w niektórych przypadkach konieczna jest zmiana parametrów algorytmów takich jak: rodzaj falki czy rząd dekompozycji.
2. Najlepsze rezultaty otrzymano stosując metody wieloskalowej detekcji krawędzi.
3. Dużą zaletą metody Row-Column jest możliwość decydowania o rodzaju wykorzystanej falki. Dzięki temu uzyskano poprawną detekcję krawędzi dla obrazów zaszumionych stosując falkę db4, jak i dla obrazów, w których krawędź nie odróżnia się wyraźnie od tła (stosując falkę rbio3.1)
4. Mimo zadowalających wyników nie jest zalecane automatyczne przekazywanie uzyskanych wyników do dalszej analizy. Za każdym razem użytkownik powinien zweryfikować otrzymane rezultaty.
5. Znacznie lepsze rezultaty osiągnięto dla obrazów niezaszumionych, dlatego też zalecana jest wstępna obróbka obrazu w postaci odszumiania.

Literatura:

1. Jacek W. Hennel, Zbigniew Olejniczak *„Jak zrozumieć falki. Podstawy falkowej analizy sygnałów”*, Kraków, 2010
2. Jan T. Białasiewicz *„Falki i aproksymacje”*, Warszawa, 2004
3. Jian-Jiun Ding *„Time Frequency Analysis and Wavelet Transforms”*, National Taiwan University, Taiwan, 2009
4. Stephane Mallat *„A Wavelet Tour of Signal Processing”*, Academic Press , 1999
5. Stephane Mallat, Nicolas Treil, Ruzena Bascy *„Image Wavelet Decomposition and Applications”*, University of Pennsylvania, Philadelphia, PA, USA, 1989
6. Jun Li *„A Wavelet Approach to Edge Detection”*, Sam Houston State University, Huntsville, TX, USA 2003
   http://www.shsu.edu/~mth_jxw/pdfflies/thesisJunLi.pdf
7. Yahia S. Al-Halabi, Hesgam Jondi Abd, *„New Wavelet-based techniques for edge detection”*, Journal of Theoretical and Applied Information Technology, 2009
   http://www.jatit.org/volumes/research-papers/Vol23No1/5Vol23No1.pdf
8. J. Petrova, E. Hostalkova *“Edge detection in medical images using the wavelet transform”*, Department of Computing and Control Engineering, Institute of Chemical Technology, Praga, Czechy
   http://dsp.vscht.cz/konference_matlab/MATLAB10/full_text/078_Petrova.pdf
9. Shehrzad Qureshi, *„Embedded Image Processing on the TMS320C6000™ DSP”* Palo Alto, CA, USA 2005
10. http://en.wikipedia.org/wiki/Sobel_operator