Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Informatyka

Rozprawa doktorska

Adaptacyjny system wspomagający usuwanie nadsegmentacji w obrazach poddanych transformacji wododziałowej

Jakub Smołka

Promotor: dr hab. Stanisław Grzegórski, prof. Politechniki Lubelskiej

Gliwice, 2010

Spis treści

Spis treści **i**

1 Wstęp **1**

2 Transformacja wododziałowa oraz problem nadsegmentacji **4**
2.1 Wprowadzenie 4
2.2 Sformułowanie transformacji wododziałowej 6
2.2.1 Transformacja wododziałowa przez zanurzanie 6
2.2.2 Transformacja wododziałowa oparta na odległości topograficznej 7
2.2.3 Transformacja wododziałowa z wykorzystaniem warunku lokalnego 8
2.3 Podział algorytmów realizujących transformację wododziałową . . 9
2.4 Algorytm transformacji wododziałowej z kategorii symulacji opadów deszczu oparty na zmodyfikowanym warunku lokalnym 13
2.5 Przygotowanie danych wejściowych dla transformacji wododziałowej 19
2.5.1 Obliczanie obrazu gradientowego dla obrazów monochromatycznych 20
2.5.2 Obliczanie obrazu gradientowego dla obrazów barwnych . . 23
2.6 Problem nadsegmentacji 26
2.7 Przegląd metod redukcji nadsegmentacji 27

3 Redukcja nadsegmentacji **49**
3.1 Wprowadzenie 49
3.2 Przygotowanie obrazu i wstępna redukcja nadsegmentacji 49
3.2.1 Zastosowanie selektywnego filtru rozmywającego 51
3.2.2 Progowanie obrazu gradientowego 53
3.2.3 Zalewanie płytkich zlewisk 54
3.3 Redukcja nadsegmentacji z zastosowaniem analizy skupień 60
3.3.1 Atrybuty zlewisk 61

3.3.2 Klasteryzacja zlewisk 65
3.3.3 Uzyskanie ostatecznej segmentacji 71

4 Dobór parametrów analizy skupień 76
4.1 Problem jakości segmentacji 76
4.2 Podział metod oceny 77
4.3 Ocena jakości segmentacji obrazów monochromatycznych 79
4.4 Dobór optymalnych parametrów analizy skupień dla obrazów monochromatycznych 82
4.5 Ocena jakości segmentacji obrazów barwnych 90
4.6 Dobór optymalnych parametrów analizy skupień dla obrazów barwnych 94

5 Poprawa wydajności metody 102
5.1 Zastosowanie transformaty falkowej 103
5.2 Transformata Haara 104
5.3 Zmniejszanie rozdzielczości obrazu za pomocą transformaty falkowej 104
5.4 Przenoszenie wyników segmentacji do oryginalnej rozdzielczości . . 105
5.5 Pełny algorytm segmentacji 109
5.6 Wyniki 110

6 Adaptacyjny system wspomagający usuwanie nadsegmentacji 116
6.1 Wprowadzenie 116
6.2 Uzyskanie nadmiernie posegmentowanego obrazu 117
6.3 Dobór parametrów klasteryzacji 117
6.4 Generowanie segmentacji o różnej liczbie klas 121
6.5 Optymalizacja liczby klas 122
6.6 Przyspieszanie za pomocą falek 124
6.7 Ręczna korekcja segmentacji 126

7 Podsumowanie 129

Bibliografia 132

Dodatki 141

A Hierarchiczne metody analizy skupień 142
A.1 Etapy analizy skupień 142
A.2 Ocena jakości analizy skupień 144
A.3 Metody grupowania 145
A.4 Współczynniki podobieństwa i miary odległości 147
A.5 Metody standaryzacji 150

Spis skrótów 152

Spis rysunków **154**

Spis tabel **156**

Rozdział 1

Wstęp

Segmentacja jest ważnym etapem przetwarzania obrazu, który pozwala na jego późniejszą analizę. Głównym jej celem jest znalezienie podziału obrazu na części, które możliwie dobrze odpowiadają rzeczywistym obiektom przedstawionym na tym obrazie [111]. Od dokładności segmentacji zależą wyniki dalszych etapów jego przetwarzania.

Jedną z metod segmentacji jest transformacja wododziałowa. Wymaga ona, aby w obrazie wejściowym wysokie wartości odpowiadały krawędziom obiektów, natomiast niskie obszarom jednorodnym. Można go traktować jak rzeźbę terenu, z górami i dolinami. Zasadę działania transformacji wododziałowej często wyjaśnia się za pomocą analogii do deszczu padającego na wspomniany teren. Woda pod wpływem grawitacji spływa po zboczach gór i zbiera się w dolinach [45]. Wszystkie punkty, z których spływa w kierunku tego samego minimum, nazywane są zlewiskiem. Omawiana transformacja znajduje w rzeźbie terenu poszczególne zlewiska. Są one rozdzielone ciągłymi liniami działów wodnych. Ciągłość krawędzi zlewisk jest istotną zaletą transformacji wododziałowej. Opisywana metoda została zaproponowana przez S. Beuchera i C. Lantuejoul w 1979 roku [15]. Od tego czasu powstały inne alternatywne sformułowania transformacji, a także wiele realizujących ją algorytmów [90]. Mimo istotnych różnic pomiędzy poszczególnymi wersjami transformacji wododziałowej ich wspólną, niestety niekorzystną, cechą, jest problem nadsegmentacji. Polega on na tym, że wynik działania tej transformacji zawiera zazwyczaj dużo więcej regionów niż rzeczywista liczba obiektów występujących w przetwarzanym obrazie. Przyczyną powstawania nadsegmentacji jest fakt, iż transformacja wododziałowa tworzy dla każdego minimum lokalnego osobne zlewisko, a w większości obrazów takich minimów jest bardzo dużo. Dodatkowo większość segmentowanych obrazów nie spełnia wymagań dotyczących obrazu wejściowego i musi być poddana działaniu filtru wykrywającego krawędzie. Filtry takie są bardzo wrażliwe nawet na niewielkie niejednorodności

obrazu i w ten sposób potęgują zjawisko nadsegmentacji. Z tego względu wszystkie metody segmentacji oparte na transformacji wododziałowej muszą zawierać rozwiązania zapewniające zmniejszenie tego zjawiska. Niestety metody redukcji nadsegmentacji są, według ich twórców, metodami uniwersalnymi albo bardzo precyzyjnie dostosowanymi do konkretnego zadania. Metody pierwszego rodzaju nie gwarantują uzyskania optymalnych wyników w każdym przypadku, natomiast metody drugiego rodzaju charakteryzuje brak możliwości łatwej adaptacji do innego konkretnego zadania. Dlatego za cel niniejszej pracy przyjęto **stworzenie adaptacyjnego systemu umożliwiającego redukcję nadsegmentacji, przy użyciu metody opartej na analizie skupień**. System ten powinien mieć możliwość szybkiej zmiany kryteriów redukcji nadsegmentacji.

Dla tak sformułowanego celu postawiono następujące tezy:

1. **Za pomocą metody opartej na analizie skupień można skutecznie zmniejszać nadsegmentację w obrazach poddanych transformacji wododziałowej.**

2. **Proces segmentacji obrazu, za pomocą proponowanego rozwiązania, można przyspieszyć przy użyciu transformaty falkowej.**

3. **Metoda redukcji nadsegmentacji oparta na analizie skupień może być dostosowana do różnych zadań.**

Niniejsza praca składa się ze wstępu, pięciu głównych rozdziałów oraz podsumowania uzyskanych wyników. Poniżej przedstawiono krótką charakterystykę głównych rozdziałów pracy.

Rozdział 2 w całości poświęcono omówieniu transformacji wododziałowej [15, 90] oraz związanemu z nią zjawisku nadsegmentacji. Przedstawiono w nim trzy sformułowania transformacji oraz scharakteryzowano algorytmy, które ją realizują. Wskazano powiązania pomiędzy definicjami a algorytmami. Ze względu na mnogość dostępnych algorytmów, szczegółowo opisano algorytm transformacji wododziałowej, którego używano we wszystkich badaniach przedstawionych w niniejszej pracy. Omówiono dostępne, w tym zastosowane w niniejszej pracy do przygotowania obrazu, filtry gradientowe. Przedstawiono wersje przeznaczone do użycia z obrazami monochromatycznymi oraz barwnymi. W ostatniej części rozdziału scharakteryzowano problem nadsegmentacji oraz dokonano przeglądu istniejących metod redukcji nadsegmentacji. Przegląd został oparty na zaproponowanym przez autora podziale tych metod.

W rozdziale 3 przedstawiono opracowaną przez autora metodę redukcji nadsegmentacji. Składa się ona z dwóch etapów. Etap pierwszy czyli wstępna redukcja nadsegmentacji odbywa się w trzech krokach: (1) zastosowanie selektywnego filtru rozmywającego [80, 93, 123], (2) progowanie obrazu gradientowego [45], (3) zalewanie płytkich zlewisk [25, 129]. Tego typu redukcja zapewnia częściowe

ale zazwyczaj niewystarczające zmniejszenie liczby zlewisk. W celu sprowadzenia liczby klas do odpowiedniego poziomu dodany został drugi etap – zasadnicza redukcja nadsegmentacji przeprowadzona w oparciu o metody hierarchicznej analizy skupień. W rozdziale 3 zostały szczegółowo przedstawione algorytmy redukcji nadsegmentacji. Opisano również wykorzystywane atrybuty zlewisk, a także sposób tworzenia segmentacji obrazu na podstawie hierarchii podobieństwa zlewisk. Metody analizy skupień, które znalazły zastosowanie w opracowanej metodzie, opisano szczegółowo w dodatku A.

Rozdział 4 dotyczy problemu doboru optymalnych parametrów analizy skupień. Prezentuje on wyniki przeprowadzonego eksperymentu, w trakcie którego: (1) tworzono segmentacje różnych typów obrazów, (2) dla uzyskanych segmentacji obliczano funkcje oceny, (3) wybrano parametry pozwalające uzyskać największą liczbę dobrych segmentacji. Doboru parametrów dokonano dla obrazów monochromatycznych oraz barwnych. Do prób wykorzystano monochromatyczne obrazy CT oraz MRI, i barwne fotografie. W rozdziale przedstawiono uzyskane wyniki, przykładowe segmentacje oraz scharakteryzowano pojawiające się prawidłowości.

Uzyskanie segmentacji obrazu można przyspieszyć dzięki analizie wielorozdzielczej [48, 49]. Przyspieszenie uzyskuje się kosztem jakości segmentacji. W rozdziale 5 przedstawiono rozwiązanie, wykorzystujące transformatę falkową, które autor dostosował do zaprezentowanej wcześniej metody. Rozwiązanie to polega na uzyskaniu piramidy obrazów, przeprowadzeniu segmentacji obrazu w niskiej rozdzielczości, oraz przeniesieniu wyników do oryginalnej rozdzielczości. Warto zaznaczyć, że zmniejszenie rozdzielczości nie tylko przyspiesza uzyskanie wyników, ale także zmniejsza nadsegmentację. Rozdział 5 zawiera również przykłady pokazujące wpływ rozmiarów segmentowanego obrazu na jakość segmentacji oraz przyspieszenie, które można uzyskać.

Ostatni spośród pięciu głównych rozdziałów – czyli rozdział 6 – prezentuje opracowany system redukcji nadsegmentacji. Scharakteryzowano możliwości w zakresie doboru parametrów segmentacji obrazu (w tym możliwości doboru parametrów klasteryzacji zlewisk). Pokazano, jak dzięki nim można zaadaptować zaimplementowaną metodę do tworzenia dobrych segmentacji ze względu na inne kryteria niż funkcje oceny segmentacji przedstawione w rozdziale 4. Następnie omówiono sposób przetwarzania wyników klasteryzacji zlewisk, w tym ręczny, bądź wspomagany za pomocą funkcji oceny, dobór liczby klas w ostatecznej segmentacji. W rozdziale 6 przedstawiono również możliwości związane z wykorzystaniem transformaty falkowej do poprawy wydajności procesu przetwarzania obrazu, a także zasadę działania opcji pozwalającej na ręczną korektę segmentacji – łączenie kilku klas w jedną i rozdzielanie obiektów błędnie połączonych.

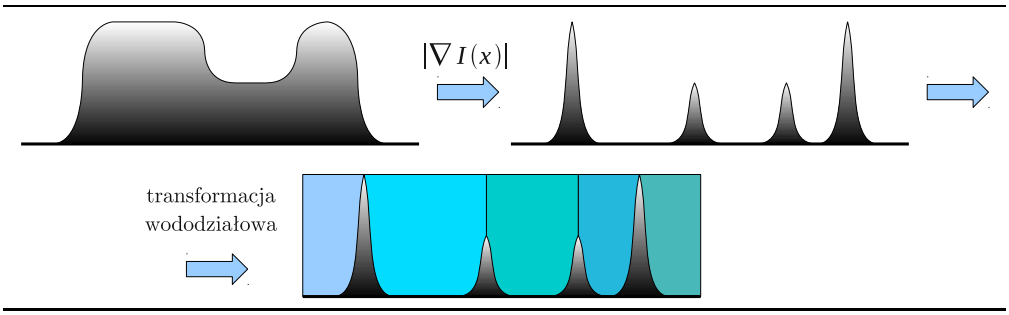
Rozdział 2

Transformacja wododziałowa oraz problem nadsegmentacji

2.1 Wprowadzenie

Transformacja wododziałowa nazywana również segmentacją wododziałową została zaproponowana przez S. Beuchera i C. Lantuejoul w 1979 roku [15]. Służy ona do segmentacji obrazów, w których wysokie wartości odpowiadają krawędziom, natomiast niskie jednorodnym obszarom. Taki obraz można uważać za odwzorowanie rzeźby terenu z górami (jasne obszary) i dolinami (ciemne obszary). Zasada działania transformacji wododziałowej jest prosta i można ją wyjaśnić używając analogii do deszczu padającego na wspomniany nierówno ukształtowany teren [15]. Padające krople deszczu pod wpływem grawitacji spływają po zboczach gór. Woda zbiera się w dolinach. Wszystkie punkty, z których woda spływa w kierunku tego samego minimum, nazywane są zlewiskiem. Obszary powstających zbiorników wodnych powiększają się wraz z ilością otrzymywanego opadu. W pewnym momencie docierają do granic zlewiska i woda z jednego zbiornika mogłaby się przelać do drugiego. W standardowej metodzie transformacji wododziałowej łączenie się obszarów jest blokowane. Obrazowo to działanie można nazwać budowaniem zapór, które razem z grzbietami łańcuchów górskich przedstawia się na obrazie jako linie działów wodnych, rozdzielające poszczególne zlewiska.

Bardziej formalnie wynikiem transformacji wododziałowej jest obraz podzielony na obszary o ciągłych krawędziach, co stanowi istotną zaletę tej metody. Natomiast jej wadą jest tworzenie zlewiska dla każdego minimum lokalnego obrazu, co przyczynia się do powstawania zjawiska nadsegmentacji, często dodatkowo potęgowanego sposobem przygotowania obrazu wejściowego. Jak już wspomniano, wymogiem transformacji wododziałowej jest, aby wysokie wartości odpowiadały



Rysunek 2.1: Ilustracja typowego sposobu obliczania transformacji wododziałowej obrazu. Obraz przedstawiono interpretując jasność pikseli jako wysokość mierzoną od pewnego poziomu odniesienia.

krawędziom segmentowanych obiektów. Wiele obrazów nie spełnia tego warunku. W celu przekształcenia ich do wymaganej postaci często stosuje się filtr obliczający moduł gradientu w poszczególnych punktach obrazu. Filtry tego rodzaju są bardzo wrażliwe na szum i inne niewielkie zmiany wartości w obrazie, co powoduje powstanie wielu minimów lokalnych a co za tym idzie – nadsegmentacji. Typowy przebieg prostej segmentacji z użyciem transformacji wododziałowej przedstawiono na rysunku 2.1.

Transformacja wododziałowa jest popularną metodą segmentacji, która znalazła wiele zastosowań. Bardzo często jest wykorzystywana do segmentacji obrazów medycznych. Są nimi obrazy CT (pochodzące z tomografu komputerowego) [1, 121, 122] jak i MRI (z rezonansu magnetycznego) [28, 43, 56, 59, 61, 83, 114, 117]. Inną kategorią segmentowanych obrazów medycznych są obrazy komórek w tym krwinek, uzyskane przy użyciu mikroskopu świetlnego [2, 47, 59, 78]. Za pomocą transformacji wododziałowej segmentowane są również obrazy pochodzące z mikroskopu elektronowego [9, 10, 12, 118]. Pojawiają się również przemysłowe zastosowania opisywanej metody, na przykład transformację wododziałową wykorzystywano w procesie oceny jakości spawów [3], do rozpoznawania sposobu obróbki powierzchni [7], a także do analizy próbek skał [75]. Innym obszarem zastosowań jest segmentacja rozmaitych obiektów w nagraniach wideo. Przykładem może być tu segmentacja przeszkód znajdujących się na drodze [10, 12, 14], małych obiektów, takich jak filmowane z dużej odległości samoloty [44], nagrań zawodów sportowych [72], a także segmentacja diagnostycznych nagrań wykonanych kamerą termowizyjną [79]. Mniej typowymi, ale również proponowanymi, zastosowaniami transformacji wododziałowej są: segmentacja tekstu [50, 51], segmentacja map odległości euklidesowej obliczonych dla obrazów binarnych [3, 117], segmentacja map odległości w obrazach stereowizyjnych [6], analiza i przetwarzanie (np. normalizacja) sygnałów jednowymiarowych [21], czy segmentacja różnych rodzajów tekstur występujących w obrazie [53].

2.2 Sformułowanie transformacji wododziałowej

Istnieją różne sposoby definiowania transformacji wododziałowej. Wielu autorów modyfikowało jej oryginalną wersję [15], a także pracowało nad zupełnie nowymi definicjami na przykład na potrzeby implementacji równoległych [19, 90]. Poniżej przedstawiono trzy podstawowe formalne definicje transformacji wododziałowej [90]: (1) transformacja wododziałowa przez zanurzanie, (2) transformacja wododziałowa oparta na odległości topograficznej, (3) transformacja wododziałowa oparta na warunku lokalnym.

2.2.1 Transformacja wododziałowa przez zanurzanie

Formalną definicję oraz algorytm realizujący transformację wododziałową przez zanurzanie zaproponowaną w [15] podali L. Vincent i P. Soille [117]. Definicja ta wykorzystuje kilka pojęć, które zostaną przedstawione poniżej.

Jeżeli A jest zbiorem punktów, to *odległość geodezyjną* $d_A(a,b)$ między punktami a i b definiuje się jako najkrótszą ścieżkę składającą się z punktów należących do zbioru A i łączącą punkty a i b [90]. Wykorzystując $d_A(a,b)$ można zdefiniować odległość pomiędzy punktem a a zbiorem $B \subset A$ [90]:

$$d_A(a,B) = \min_{b \in B} d_A(a,b). \tag{2.1}$$

Zbiór punktów B nie musi być zbiorem spójnym, ale można go podzielić na spójne podzbiory punktów B_i przy czym $i = 1 \ldots k$. Wtedy dla każdego zbioru B_i można zdefiniować jego *geodezyjną strefę wpływów* w zbiorze A [90]:

$$iz_A(B_i) = \{x \in A | \forall j \in [1 \ldots k] \backslash \{i\} : d_A(x,B_i) < d_A(x,B_j)\}. \tag{2.2}$$

Na podstawie stref wpływów wszystkich zbiorów składowych B_i można określić *strefę wpływów zbioru* $B \subset A$ w zbiorze A [90]:

$$IZ_A(B) = \bigcup_{i=1}^{k} iz_A(B_i). \tag{2.3}$$

Oprócz *strefy wpływów zbioru* do zdefiniowania transformacji wododziałowej konieczne jest określenie *zbioru* T_h zawierającego wszystkie punkty, których wartość w obrazie g nie przekracza progu h [90]:

$$T_h = \{x \in D | g(x) \leqslant h\}, \tag{2.4}$$

gdzie D to zbiór punktów obrazu.

Wykorzystując pojęcia *strefy wpływów* (wzór 2.3) i *zbiór punktów* T_h (wzór 2.4) można rekurencyjnie zdefiniować transformację wododziałową [90]:

$$\begin{cases} X_{h_{min}} & = T_{h_{min}} \\ X_{h+1} & = MIN_{h+1} \cup IZ_{T_{h+1}}(X_h),\ h \in [h_{min}, h_{max}). \end{cases} \tag{2.5}$$

Algorytm rozpoczyna pracę od znalezienia minimów o najmniejszej wysokości h_{min} występującej w obrazie g. Następnie przegląda kolejne poziomy (od h_{min} do h_{max}) i ustala strefę wpływów zbioru X_h (wyniku poprzedniej iteracji) w zbiorze T_{h+1}. Do wyznaczonej strefy wpływów dodaje zbiór minimów MIN_{h+1} o wysokości $h+1$. Gdy zostanie wyznaczony zbiór $X_{h_{max}}$, zbiór punktów tworzących linie wododziałowe $W(g)$ wyznacza się jako:

$$W(g) = D \backslash X_{h_{max}}. \tag{2.6}$$

2.2.2 Transformacja wododziałowa oparta na odległości topograficznej

Transformacja wododziałowa została zdefiniowana w oparciu o *odległość topograficzną* [68]. Odległość topograficzna obliczana jest na podstawie kosztu przejścia między dwoma sąsiednimi punktami. Koszt ten jest z kolei określany na podstawie nachylenia zbocza. *Nachylenie zbocza* w punkcie x jest określone wzorem [90]:

$$LS(x) = \max_{y \in N_G(x) \cup \{x\}} \left(\frac{g(x) - g(y)}{d(x,y)} \right), \tag{2.7}$$

gdzie $N_G(x)$ oznacza sąsiedztwo punktu x, $d(x,y)$ to odległość między punktami x i y. Ponadto dla przypadku $x = y$ ułamek w równaniu 2.7 przyjmuje się równy 0. Na podstawie nachylenia LS można zdefiniować koszt przejścia między punktami x i y. Koszt $c(x,y)$ jest określony w następujący sposób [90]:

$$c(x,y) = \begin{cases} LS(x) \cdot d(x,y) & \text{jeżeli } g(x) > g(y) \\ LS(y) \cdot d(x,y) & \text{jeżeli } g(x) < g(y) \\ 1/2 \cdot [LS(x) + LS(y)] \cdot d(x,y) & \text{jeżeli } g(x) = g(y). \end{cases} \tag{2.8}$$

Odległość topograficzna prowadząca wzdłuż ścieżki π składającej się z punktów $x = x_0, x_1, \ldots, x_t = y$ wyraża się wzorem [90]:

$$T_g^{\pi}(x,y) = \sum_{i=0}^{t-1} d(x_i, x_{i+1}) c(x_i, x_{i+1}). \tag{2.9}$$

Powyższe równanie pozwala na określenie odległości topograficznej $T_g(x,y)$ między dwoma punktami x i y jako najmniejszej odległości topograficznej wzdłuż wszystkich możliwych ścieżek $\Pi(x,y)$ łączących punkty x i y:

$$T_g(x,y) = \min_{\pi \in \Pi(x,y)} T_g^{\pi}(x,y). \tag{2.10}$$

W omawianym sformułowaniu transformacji wododziałowej zlewisko związane z minimum lokalnym m_i jest określone jako zbiór punktów których odległość topograficzna od minimum m_i jest mniejsza od jakiegokolwiek innego minimum m_j [90]:

$$CB(m_i) = \{x \in D | \forall j \in I \backslash i : f(m_i) + T_g(x, m_i) < g(m_j) + T_g(x, m_j)\}. \tag{2.11}$$

Na podstawie zlewisk możliwe jest wyznaczenie linii wododziałowych $W(g)$, które tworzą punkty obrazu nienależące do żadnego ze zlewisk:

$$W(g) = D \setminus \left(\bigcup_{i \in I} CB(m_i) \right). \tag{2.12}$$

Transformacja wododziałowa oparta na odległości topograficznej może dawać inne wyniki niż transformacja wododziałowa symulująca zanurzanie. W szczególności metoda ta częściej tworzy tak zwane *grube linie wododziałowe* czyli linie o grubości większej niż jeden punkt [90]. W specyficznych przypadkach punktów wododziałowych jest tak dużo, że bardziej trafnym określeniem byłby "obszar wododziałowy"

Dodatkowym problemem przy stosowaniu powyższego sformułowania transformacji wododziałowej jest wymóg, aby w obrazie nie występowały równiny, które nie są minimami. Rozwiązaniem tego problemu jest *uzupełnienie obrazu* (ang. *lower completion*) [90]. Polega ono na zmianie zakresu wartości obrazu (mnożenie przez wartość L_c) i modyfikacji punktów leżących na równinach w zależności od ich odległości od brzegu równiny

$$g' = \begin{cases} L_c \cdot g(x) & \text{jeżeli } d(x) = 0 \\ L_c \cdot g(x) + d(x) - 1 & \text{w przeciwnym wypadku,} \end{cases} \tag{2.13}$$

gdzie $d(x)$ to odległość punktu x do najbliższego punktu o niższej wartości y (ścieżka łącząca x i y musi być nierosnąca dlatego w przypadku punktów należących do minimum lokalnego przyjmuje się $d(x) = 0$), natomiast $L_c = \underset{x \in D}{max}\, d(x)$ [90].

2.2.3 Transformacja wododziałowa z wykorzystaniem warunku lokalnego

Trzecia definicja transformacji wododziałowej jest oparta na *warunku lokalnym*. Podobnie do poprzedniej również wykorzystuje odległość topograficzną, jednak w odróżnieniu od niej nie tworzy linii wododziałowych [90]. Zamiast tego punkty należące do poszczególnych zlewisk otrzymują unikalne etykiety. Podejście to zaproponowali A. Bieniek i in. [19, 20]. Lokalny charakter zależności zawartych w definicji ułatwia zrównoleglenie algorytmu transformacji wododziałowej. Możliwe jest podzielenie obrazu na fragmenty, nadanie punktom tymczasowych etykiet, a następnie ich ujednolicenie we wszystkich fragmentach.

Według definicji opartej na warunku lokalnym transformacją wododziałową obrazu jest każda funkcja L przypisująca punktom etykiety i spełniająca poniższe dwa warunki [90]:

1. $L(m_i) \neq L(m_j)\ \forall i \neq j$ gdzie $\{m_k\}_{k \in I}$ to zbiór minimów obrazu g,
2. dla każdego piksela x spełniającego warunek $\Gamma(x) \neq \emptyset$ istnieje $x' \in \Gamma(x)$ takie że $L(x) = L(x')$,

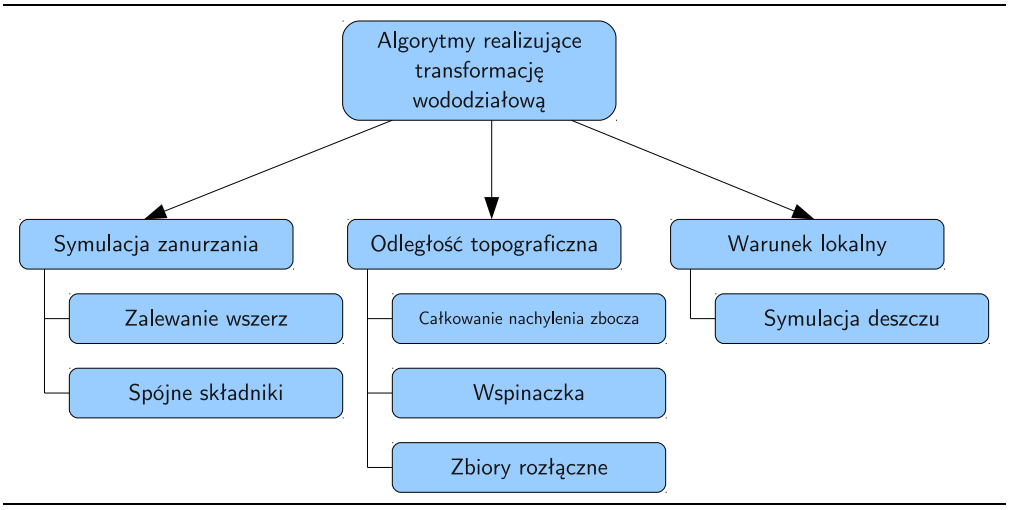
przy czym $\Gamma(x)$ to zbiór punktów, taki że dla wszystkich $y \in \Gamma(x)$ spełniony jest warunek $\frac{g(x)-g(y)}{d(x,y)} = LS(x)$ [90].

2.3 Podział algorytmów realizujących transformację wododziałową

Omówione w punkcie 2.2 definicje stanowią punkt wyjścia do opracowania algorytmów realizujących transformację wododziałową. Definicje stanowią też kryterium, według którego można dokonać podziału tych implementacji. Rysunek 2.2 przedstawia taką klasyfikację. Warto również zaznaczyć, że niektóre algorytmy nie zachowują całkowitej zgodności z definicją. Odstępstwa mają zazwyczaj na celu poprawę złożoności obliczeniowej algorytmu [90].

Najpopularniejszym algorytmem realizującym transformację wododziałową przez zanurzanie jest algorytm zaproponowany przez L. Vincenta i P. Soille w [117]. Jego działanie można przedstawić jako zanurzanie rzeźby terenu w wodzie. W zanurzanej powierzchni znajdują się dziury, w miejscach odpowiadających minimom lokalnym. Woda przedostaje się przez dziury i podczas zanurzania zalewa kolejne poziomy terenu. Aby nie dopuścić do połączenia się wody, z obszarów odpowiadających różnym minimom budowane są tamy. Algorytm składa się z dwóch etapów [90]. Pierwszym z nich jest sortowanie pikseli pod względem ich poziomu szarości. Umożliwia to bezpośredni, a co za tym idzie szybki dostęp do współrzędnych punktów o zadanej wartości. Drugim etapem jest *zalewanie wszerz* rzeźby terenu poziom po poziomie. Algorytm jest podobny do grafowego algorytmu BFS. W kolejnych iteracjach: (a) wszystkie piksele na bieżącym poziomie (o wysokości h) otrzymują etykietę M, (b) bieżący poziom (złożony z pikseli o etykiecie M) zalewany jest wszerz, poprzez nadawanie pikselom etykiet ich sąsiadów położonych na tej samej lub niższej wysokości (algorytm wykorzystuje kolejki FIFO do propagacji etykiet). Piksele, które mają sąsiadów o różnych etykietach, otrzymują etykietę W – czyli stają się częścią linii wododziałowych, (c) piksele, które nie otrzymały nowej etykiety (tzn. w dalszym ciągu mają etykietę M) są minimami lokalnymi o wysokości h. Otrzymują one nowe etykiety i stanowią zaczątek nowych zlewisk.

Duża popularność [24, 34, 84, 89, 97, 117, 118] algorytmu symulującego zanurzanie spowodowała, że bardzo często był on modyfikowany i poprawiany. W [118] zaproponowano wersję, która umożliwia regulację kroku z jakim odbywa się zalewanie rzeźby terenu. Umożliwiło to zmniejszenie nadsegmentacji (więcej w punkcie 2.7). W [34] zajęto się wymaganiami pamięciowymi algorytmu – zaproponowano wersję, której wymagania pamięciowe zależą wyłącznie od rozmiarów obrazu a nie od jego zawartości, która może być trudna do przewidzenia. W [84] przedstawiono algorytm o poprawionej złożoności obliczeniowej, który także tworzy ciągłe i cienkie linie działów wodnych. Powstała też modyfikacja dostosowana do



Rysunek 2.2: Główne kategorie algorytmów realizujących transformację wododziałową (na podstawie [90]).

pracy na obrazie gradientowym o większej rozdzielczości niż obraz oryginalny (informacje na temat tego rodzaju gradientu umieszczono w punkcie 2.5.1) [50, 51]. Może ona zostać przedstawiona poprzez analogię do systemu krzyżujących się rur umieszczonych tuż nad powierzchnią ziemi i połączonych tylko w miejscach odpowiadających pikselom obrazu oryginalnego. Tak jak w przypadku zwykłego algorytmu woda dostaje się do systemu w miejscach minimów lokalnych. Różnica w stosunku do tradycyjnego algorytmu polega na tym, że linie wododziałowe mogą powstawać tylko w miejscach, w których istnieją połączenia w systemie rur.

Drugim wariantem algorytmu realizującego transformację wododziałową przez zanurzanie jest rozwiązanie wykorzystujące *spójne składniki* [90]. Polega ono na znalezieniu w obrazie spójnych składników, czyli spójnych obszarów o stałej wartości pikseli i opisaniu obrazu jako grafu, w którym węzłami są poszczególne składniki a krawędzie łączą sąsiadujące składniki ze sobą. Następnym krokiem jest wykonanie symulacji zalewania na grafie za pomocą uproszczonej wersji algorytmu Vincenta i Soille. Ostatnim krokiem jest przeniesienie wyników z grafu na piksele obrazu. Algorytm ten nie wymaga kolejek FIFO (przekształcenie obrazu w graf rozwiązuje problem równin) i ułatwia implementację równoległą [90].

Drugą kategorią są algorytmy wykorzystujące odległość topograficzną [90]. Pierwsze rozwiązanie tego typu oparte jest na algorytmie najkrótszych ścieżek Dijkstry-Moore'a. Jego działanie może być interpretowane jako *całkowanie nachylenia zbocza* w obrazie gradientowym. Wymagane jest wcześniejsze usunięcie równin, które nie są minimami (punkt 2.2.2). Algorytm traktuje obraz jak graf, w którym piksele są węzłami, a krawędzie posiadają wagi odpowiadające kosztom przejścia pomiędzy łączonymi pikselami. Tworzony jest również obraz pomocniczy d, w którym zapisywane są odległości poszczególnych pikseli od najbliższych

minimów lokalnych (początkowo wszystkie piksele w tym obrazie mają bardzo dużą wartość symbolizującą nieskończoność). Na początku pracy wszystkim minimom lokalnym nadawane są unikalne etykiety, a w obrazie d zapamiętywane ich wartości. Następnie znajdowany jest nieprzetworzony punkt u o najmniejszej odległości zapisanej w obrazie pomocniczym. Po czym następuje przeglądanie otoczenia punktu u, podczas którego aktualizuje się odległości i etykiet sąsiadów. W trakcie aktualizacji etykieta sąsiada v jest zmieniana na etykietę punktu u, jeżeli odległość punktu v od minimum lokalnego jest większa niż suma odległości u i kosztu przejścia z u do v. Równocześnie zapisywana jest nowa, mniejsza odległość punktu v. W przypadku gdy wspomniane wartości są równe, punkt v staje się częścią linii wododziałowej i otrzymuje etykietę W.

Innym algorytmem całkującym nachylenie zbocza jest algorytm, którego działanie polega na przeglądaniu obrazu wiersz po wierszu na przemian w przód i wstecz. Podczas tego procesu dla bieżącego piksela u wykonywana jest propagacja etykiet. Algorytm pracuje tak długo, jak długo przeglądanie obrazu i towarzysząca mu propagacja etykiet powoduje zmianę wyniku. Inicjalizacja algorytmu a także propagacja etykiet jest analogiczna do rozwiązania opisanego powyżej. Ponadto tak samo jak poprzedni, również i ten algorytm wymaga usunięcia równin.

Kolejnym algorytmem zaliczanym do omawianej kategorii jest rozwiązanie polegające na nadawaniu etykiet pikselom, w sposób przypominający *wspinaczkę* [90]. Również w tym wypadku konieczna jest eliminacja równin, ale jego zaletą jest to, że nie wymaga obliczania kosztów przejścia między pikselami. Pierwszym krokiem algorytmu jest nadanie etykiet minimom lokalnym i utworzenie zbioru S zawierającego nieprzetworzone piksele (na początku są w nim wszystkie piksele z wyjątkiem wnętrza minimów). Następnie ze zbioru S wybierany jest punkt p o najmniejszej wartości. Najwyżsi sąsiedzi punktu p, należący do zbioru S i nieposiadający jeszcze etykiety, otrzymują taką samą etykietę jak p. Jeżeli najwyższy sąsiad posiada już etykietę, to staje się punktem leżącym na linii wododziałowej i otrzymuje etykietę W.

Ostatnim ważnym wariantem algorytmów transformacji wododziałowej wykorzystującym odległość topograficzną jest rozwiązanie wykorzystujące *zbiory rozłączne* [90]. Po usunięciu równin dla otrzymanego obrazu budowany jest graf, w którym dla każdego piksela (w każdym węźle) przechowywane są wskaźniki do jego kolejnych niżej położonych sąsiadów. Dla poszczególnych minimów wybierani są reprezentanci. Wskaźniki reprezentantów wskazują na nich samych. Po zbudowaniu grafu etykietowane są minima, a następnie algorytm przegląda wszystkie piksele. Podczas przeglądania dla bieżącego piksela znajdowany jest reprezentant. W tym celu algorytm podąża z bieżącego punktu, opadającą ścieżką, do jednego z minimów lokalnych, wykorzystując przy tym utworzony wcześniej graf. Jeżeli wszystkie takie ścieżki prowadzą to tego samego minimum, to punkt otrzymuje jego etykietę. W przeciwnym wypadku staje się on częścią linii wododziałowej i

otrzymuje etykietę W. Należy dodać, że podczas wyszukiwania reprezentantów stosowana jest technika *kompresji ścieżek*, która znacznie przyspiesza wyszukiwanie reprezentantów kolejnych punktów.

Oprócz algorytmów sekwencyjnych prezentowane są również rozwiązania przeznaczone do wykonania na komputerach równoległych [19, 20, 71, 70, 90]. Jedna z pierwszych propozycji [19, 20] wyznacza transformację wododziałową w oparciu o warunek lokalny (punkt 2.2.3), a więc nie tworzy linii wododziałowych. Jest to rozwiązanie podobne do opisanego powyżej – wykorzystującego zbiory rozłączne. Jednak w odróżnieniu od niego, w wypadku gdy piksel ma kilku najniższych sąsiadów, algorytm wybiera arbitralnie jednego z nich. Ta wersja transformacji wododziałowej określana jest *symulacją opadów deszczu* [90].

Podejmowane były również próby zrównoleglenia algorytmu nadającego etykiety w sposób przypominający wspinaczkę, algorytmu przeglądającego obraz wiersz po wierszu, algorytmu opartego na grafie spójnych składników, a także rozwiązania wykorzystującego zbiory rozłączne [90].

Większość algorytmów realizujących transformację wododziałową jest przystosowana do obrazów dwuwymiarowych ponieważ ich interpretacja oraz opis działania są łatwe. Powstały również wersje służące do segmentacji trójwymiarowych obrazów złożonych z wokseli [97, 118], a także obrazów o dowolnej liczbie wymiarów [45, 117]. Transformację wododziałową uogólniono nawet na nieregularne struktury takie jak grafy zlewisk [121, 122], a także trójwymiarowe siatki złożone z trójkątów [85].

Oprócz algorytmów opisanych powyżej powstają bardziej nietypowe rozwiązania. Przykładem może być propozycja wykorzystująca kody łańcuchowe [113]. W zaproponowanym algorytmie, dla każdego piksela segmentowanego obrazu zapamiętywana jest informacja, w postaci odpowiedniego kodu, gdzie znajduje się jego najniższy sąsiad, a także, dla których sąsiadujących z nim pikseli jest on najniższym sąsiadem. Te informacje pozwalają na zidentyfikowanie minimów, nadanie im etykiet, a następnie skopiowanie tych etykiet do innych punktów należących do tych samych zlewisk.

Całkowicie odmiennym podejściem do transformacji wododziałowej jest przedstawienie jej jako problemu minimalizacji energii [73]. Składa się on z dwóch etapów. Pierwszym jest przeprowadzenie tradycyjnej transformacji wododziałowej. Krawędzie otrzymanych regionów mogą być bardzo poszarpane. Otrzymany wynik służy do inicjalizacji drugiego etapu – minimalizacji energii linii wododziałowych. "Energia" to suma odległości topograficznych punktów leżących na brzegu działów wodnych, od odpowiadających im minimów, wysokości tych minimów i długości konturów działów wodnych [73]. Drugi etap rozpoczyna się od obliczenia, dla wszystkich punktów na brzegach regionów, jaką zmianę energii spowodowałoby ich przeniesienie do sąsiedniego regionu. Następnie punkt, dla którego wyznaczono największy spadek energii, jest znajdowany i przenoszony.

Cały proces jest powtarzany do momentu, w którym nie można już zmniejszyć energii linii wododziałowych. W celu uzyskania gładkich krawędzi minimalizację energii przeprowadza się w wielu rozdzielczościach (od najmniejszej do największej).

2.4 Algorytm transformacji wododziałowej z kategorii symulacji opadów deszczu oparty na zmodyfikowanym warunku lokalnym

W poprzednim punkcie omówiono trzy główne grupy algorytmów, wyznaczających transformację wododziałową łącznie z ich zmodyfikowanymi wersjami. W wielu wypadkach różnice między nimi powodują otrzymanie niejednakowych wyników dla tego samego obrazu wejściowego [90]. Z tego względu poniżej przedstawiono szczegółowy opis algorytmu, który wykorzystano do uzyskania wszystkich segmentacji zamieszczonych w niniejszej pracy. Należy on do kategorii algorytmów, które nie tworzą linii wododziałowych i jest podobny do rozwiązań opartych na warunku lokalnym. Realizuje definicję mówiącą, że punkt należy do zlewiska związanego z minimum lokalnym, do którego prowadzi ścieżka najszybszego spadku [45]. Jeżeli w danym punkcie istnieje kilka ścieżek o takim samym – maksymalnym spadku – to jedna z nich jest arbitralnie wybierana. Dlatego algorytm można zaliczyć do grupy symulującej opady deszczu [90]. Dzięki zastosowaniu *4-sąsiedztwa* określanie nachylenia zbocza $LS(x)$ jest uproszczone (odległości między sąsiadami wynoszą 1) i sprowadza się do obliczania różnicy wartości pikseli. Aby uniknąć uzupełniania obrazu, każda równina, niebędąca minimum otrzymuje w ostatecznej segmentacji etykietę taką samą, jak jej najniższy sąsiad. Na tym polega odstępstwo od klasycznego warunku lokalnego. Jak w przypadku innych typowych algorytmów realizujących transformację wododziałową dla każdego minimum lokalnego tworzone jest osobne zlewisko, a łączenie punktów z minimami odbywa się poprzez nadawanie im tych samych etykiet.

Algorytm można podzielić na kilka etapów: (1) ustawienie wartości początkowych zmiennych i zdefiniowanie oznaczeń, (2) nadanie unikalnych etykiet poszczególnym minimom lokalnym i równinom, (3) znalezienie najniższych sąsiadów poszczególnych równin, (4) nadanie odpowiednich etykiet pozostałym pikselom (łączenie pozostałych pikseli z minimami), (5) połączenie równin z odpowiednimi minimami. W tabeli 2.1 został zamieszczony spis użytych oznaczeń wraz z objaśnieniami.

Tabela 2.1: Zestawienie oznaczeń (obiekty, klasy, metody) użytych w algorytmie analizy skupień

Metoda/pole	**Opis**
obiekt: obrazWe – przechowuje obraz wejściowy (zazwyczaj obraz gradientowy), w którym wysokie wartości odpowiadają krawędziom obiektów	
`budujMur(`*`wysokosc`*`)`	Powiększa obraz o 1 piksel z każdej strony. W rozszerzonym obszarze buduje mur zapobiegający "wylewaniu" się wody poza obszar obrazu wejściowego. Punkty obrazu wejściowego zachowują swoje współrzędne np. punkt (0,0) jest w dalszym ciągu (0,0).
`maksimum()`	Zwraca wartość globalnego maksimum występującego w obrazie.
`minimum()`	Zwraca wartość globalnego minimum występującego w obrazie.
`operator[`*`wsp`*`]`	Zwraca wartość piksela o podanych współrzędnych.
`progowanie(`*`prog`*`)`	Zastępuje wartością progu wszystkie wartości mniejsze lub równe wartości progu.
obiekt: obrazWy – przechowuje obraz wyjściowy – wynik segmentacji – zawierający etykiety zlewisk	
`operator[`*`wsp`*`]`	Zwraca etykietę piksela o podanych współrzędnych.
`ujednolicenie-Etykiet(`*`tabela`*`)`	Zamienia etykiety pikseli na najmniejszą z równoważnych etykiet. Zmiany dokonywane są na podstawie tabeli równoważnych etykiet.
`ustaw(`*`wartosc`*`)`	Zmienia wartość wszystkich pikseli na podaną wartość.
klasa: Rownina – służy do przechowywania informacji o najniższym sąsiedzie równiny	
`najnizszySasiadWart`	Wartość piksela, który jest najniższym sąsiadem równiny.
`sasiad`	Położenie najniższego sąsiada równiny.
`wysokosc`	Wysokość, na której znajduje się równina.
obiekt: rowniny – przechowuje listę równin wraz z informacjami o nich	
`aktualizuj(`*`etykieta, wartoscMinSasiada, polazenieMinSasiada`*`)`	Aktualizuje informacje o równinie posiadającej zadaną etykietę.

ciąg dalszy na następnej stronie...

Tabela 2.1 – ciąg dalszy

Metoda/pole	**Opis**
`dodaj(`*`etykieta, wartoscMinSasiada, polazenieMinSasiada`*`)`	Dodaje do spisu równinę o zadanej etykiecie. Zapamiętywana jest wartość najniższego sąsiada i jego położenie.
`polacz(`*`tabela`*`)`	Łączy stykające się płaskie regiony na podstawie tabeli równoważnych etykiet. Każda grupa regionów po połączeniu ma najmniejszą z równoważnych etykiet. W połączonym regionie zapisywany jest najmniejszy spośród minimalnych sąsiadów.
obiekt: sciezka – przechowuje ścieżkę najszybszego spadku z bieżącego punktu do minimum lokalnego	
`pop()`	Zwraca współrzędne punktu zdjęte ze stosu.
`push(`*`wspolrzedne`*`)`	Odkłada na stos współrzędne punktu.
`wyczysc()`	Czyści stos.
obiekt: tabelaRownowaznosci – przechowuje spis równoważnych etykiet (reprezentujących to samo zlewisko)	
`dodaj(`*`etykieta1, etykieta2`*`)`	Dodaje parę równoważnych etykiet do tabeli. Para etykiet jest nieuporządkowana.
`wyczysc()`	Kasuje wszystkie wpisy w tabeli.

Jak wspomniano powyżej pierwszym etapem zastosowanego algorytmu transformacji wododziałowej jest inicjalizacja (algorytm 2.1). Podczas tego etapu wszystkim pikselom obrazu wyjściowego przypisywana jest specjalna etykieta *BRAK_ETYKIETY*, która oznacza, że dany piksel nie został jeszcze przypisany do żadnego zlewiska. Znajdowane są minimalne i maksymalne wartości występujące w obrazie wejściowym (gradientowym), które będą wykorzystywane w kolejnych krokach procesu segmentacji. Na koniec budowany jest "mur" dookoła segmentowanego obrazu. Obraz powiększany jest o jeden piksel z każdej strony, a następnie w nowym obszarze ustawiana jest wartość $maksimum + 1$. Dzięki temu algorytm znajdujący ścieżki najszybszego spadku nie musi uwzględniać przypadku pracy na brzegu obrazu.

Algorytm 2.1 Definicje początkowe / inicjalizacja zmiennych

1: #define BRAK_ETYKIETY 0
2: $biezacaEtykieta \leftarrow 1$
3: $obrazWy$.ustaw(BRAK_ETYKIETY)
4: $minimum \leftarrow obrazWe$.minimum()
5: $maksimum \leftarrow obrazWe$.maksimum()
6: $obrazWe$.budujMur($maksimum + 1$)

Następnym etapem jest nadanie etykiet wszystkim minimom lokalnym i równinom (algorytm 2.2). Przeglądane są wszystkie punkty obrazu. Jeżeli bieżący punkt x nie ma etykiety, to jego otoczenie jest przeglądane. Algorytm sprawdza, czy przynajmniej jeden z jego sąsiadów ma taką samą wartość – co oznacza, że znaleziono równinę. Dodatkowo sprawdza, czy punkt x ma sąsiada o niższej wartości – co oznacza, że nie znaleziono minimum. Trzecią możliwością jest to, że bieżący piksel jest minimum lokalnym składającym się z jednego piksela. Rozróżnienie między równinami, które są i tymi, które nie są minimami lokalnymi (złożonymi z wielu punktów) nie jest istotne. W przypadku, gdy bieżący punkt x jest częścią równiny, otrzymuje etykietę sąsiada (gdy sąsiad miał już etykietę) lub nadawana jest mu nowa etykieta (gdy sąsiad również nie posiada etykiety). Nowe równiny są dodawane do spisu razem z informacją o położeniu i wartości najniższego sąsiada (na tym etapie ustawiane są tylko wartości początkowe). Następnie ponownie przeglądane jest otoczenie bieżącego punktu w poszukiwaniu punktów o tej samej wartości ale innej etykiecie. Punkty spełniające ten warunek są dodawane do tabeli równoważnych etykiet, która umożliwia połączenie fragmentów równin. Przypadek, gdy bieżący punkt x jest jednopunktowym minimum lokalnym, jest mniej skomplikowany. Taki punkt otrzymuje wyłącznie nową etykietę. Końcowymi czynnościami wykonywanymi na tym etapie jest ujednolicenie wpisów w spisie równin (wpisy są łączone na podstawie tabeli równoważnych etykiet), ujednolicenie etykiet w obrazie wyjściowym oraz wyczyszczenie tabeli równoważnych etykiet.

Po zakończeniu tego etapu algorytmu wszystkie minima lokalne (złożone z jednego lub wielu punktów) a także równiny, które nie są minimami, posiadają jednolite etykiety. Ponadto istnieje spis wszystkich równin, który zawiera wartości początkowe potrzebne do znalezienia najniższych sąsiadów tych równin.

W kolejnym etapie (algorytm 2.3) ponownie przeglądane są wszystkie punkty obrazu. Jeżeli znaleziono punkt posiadający etykietę, to przeglądane jest jego otoczenie. Jeżeli jeden z sąsiadów ma mniejszą wartość niż punkt bieżący x oznacza to, że punkt x, należy do równiny, która nie jest minimum lokalnym i należy zaktualizować dane o najniższym sąsiedzie bieżącej równiny (funkcja *aktualizuj()* modyfikuje dane, wtedy gdy wartość sąsiada jest mniejsza niż znaleziona do tej pory). Dodatkowo otoczenie bieżącego punktu jest sprawdzane pod kątem punktów o tej samej wartości ale o innej etykiecie. Jeżeli takie punkty są znalezione, to ich etykiety dodawane są do tabeli równoważnych etykiet. Pod koniec tego etapu ponownie łączone są fragmenty równin i zmieniane są etykiety w obrazie wyjściowym. Do tych czynności wykorzystywana jest tabela równoważności, która następnie jest czyszczona.

Po zakończeniu tego etapu spis równin zawiera już właściwe wartości i współrzędne najniższych sąsiadów wszystkich równin, co umożliwi ustalenie minimów, z którymi należy je połączyć.

Algorytm 2.2 Nadanie etykiet minimom lokalnym i równinom

```
1: for all x ∈ D do
2:   if obrazWy[x] = BRAK_ETYKIETY then
3:     minimum1Px ← true; rownina ← false
4:     for all s ∈ N(x) do
5:       if obrazWe[x] = obrazWe[s] then
6:         rownina ← true; sasiad ← s; break
7:       else if obrazWe[x] > obrazWe[s] then
8:         minimum1Px ← false
9:       end if
10:    end for
11:    if rownina then
12:      if obrazWy[sasiad] ≠ BRAK_ETYKIETY then
13:        obrazWy[x] ← obrazWy[sasiad]
14:      else
15:        obrazWy[x] ← biezacaEtykieta
16:        rowniny.dodaj(biezacaEtykieta,maksimum + 1,x)
17:        biezacaEtykieta ← biezacaEtykieta + 1
18:      end if
19:      for all s ∈ N(x) do
20:        if obrazWe[s] = obrazWe[x] and
           obrazWy[s] ≠ BRAK_ETYKIETY and
           obrazWy[s] ≠ obrazWy[x] then
21:          tabelaRownowaznosci.dodaj(obrazWy[s], obrazWy[x])
22:        end if
23:      end for
24:    else if minimum1Px then
25:      obrazWy[x] ← biezacaEtykieta
26:      biezacaEtykieta ← biezacaEtykieta + 1
27:    end if
28:  end if
29: end for
30: rowniny.polacz(tabelaRownowaznosci)
31: obrazWy.ujednolicenieEtykiet(tabelaRownowaznosci)
32: tabelaRownowaznosci.wyczysc()
```

Algorytm 2.3 Znajdowanie najniższych sąsiadów równin

```
1: for all x ∈ D do
2:   if obrazWy[x] ≠ BRAK_ETYKIETY then
3:     for all s ∈ N(x) do
4:       if obrazWe[x] > obrazWe[s] and
         rowniny[obrazWy[x]].najnizszySasiadWart > obrazWe[x] then
5:         rowniny.aktualizuj(obrazWy[x],obrazWe[s],s);
6:       end if
7:       if obrazWe[x] = obrazWe[s] and obrazWy[x] ≠ obrazWy[s] then
8:         tabelaRownowaznosci.dodaj(obrazWy[s],obrazWy[x])
9:       end if
10:    end for
11:  end if
12: end for
13: rowniny.polacz(tabelaRownowaznosci)
14: obrazWy.ujednolicenieEtykiet(tabelaRownowaznosci)
15: tabelaRownowaznosci.wyczysc()
```

Przedostatnim etapem algorytmu realizującego transformację wododziałową jest powiązanie pikseli nieposiadających etykiet z odpowiednimi minimami (algorytm 2.4). Ponownie przeglądane są wszystkie punkty obrazu. Jeżeli algorytm natrafi na punkt bez etykiety, to dodaje jego współrzędne do ścieżki i przegląda jego otoczenie w poszukiwaniu sąsiada o najniższej wartości. Warto zaznaczyć, że na tym etapie wszystkie punkty bez etykiety posiadają niższych sąsiadów (wszystkim pozostałym już nadano etykiety), a więc nie ma konieczności sprawdzania, czy udało się znaleźć sąsiada o niższej wartości. Algorytm przechodzi do znalezionego punktu (porusza się ścieżką najszybszego spadku). Jeżeli nowy punkt bieżący również nie ma etykiety, to jest dodawany do ścieżki i poszukiwanie najniższego sąsiada rozpoczyna się od nowa. W pewnym momencie algorytm musi napotkać punkt posiadający już etykietę. Wtedy odczytuje kolejne punkty ze ścieżki najszybszego spadku i nadaje im znalezioną etykietę.

Po tym etapie wszystkie punkty posiadają etykiety, takie same jak minima lokalne lub równiny niebędące minimami, które leżą na przechodzącej przez nie ścieżce najszybszego spadku.

Ostatni etap ma za zadanie powiązanie równin z odpowiednimi minimami (algorytm 2.5). Polega on na przejrzeniu wszystkich równin zapisanych w spisie. Dla każdej równiny odczytywane jest położenie oraz etykieta jej najniższego sąsiada. Do tabeli równoważności dodawane są pary składające się z etykiet równin i etykiet ich najniższych sąsiadów. Ostatnim krokiem jest ujednolicenie etykiet w obrazie wyjściowym.

Po zakończeniu tego etapu obraz wyjściowy zawiera wynik transformacji wododziałowej obrazu wejściowego (gradientowego).

Algorytm 2.4 Łączenie pozostałych pikseli z minimami lokalnymi

```
1: sciezka.wyczysc()
2: for all x ∈ D do
3:   if obrazWy[x] = BRAK_ETYKIETY then
4:     xRob ← x
5:     while obrazWy[xRob] = BRAK_ETYKIETY do
6:       sciezka.push(xRob); min ← obrazWe[xRob]
7:       for all s ∈ N(xRob) do
8:         if min > obrazWe[s] then
9:           xRob ← s; min ← obrazWe[s]
10:        end if
11:      end for
12:    end while
13:    while not sciezka.pusty() do
14:      xRob2 ← sciezka.pop()
15:      obrazWy[xRob2] ← obrazWy[xRob]
16:    end while
17:  end if
18: end for
```

Algorytm 2.5 Łączenie równin z odpowiadającymi im minimami

```
1: for pr = 0 to biezacaEtykieta do
2:   if rowniny[pr] ≠ NULL and
     rowniny[pr].najnizszySasiadWart < maksimum + 1 then
3:     tabelaRownowaznosci.dodaj(pr,obrazWy[rowniny[pr].sasiad])
4:   end if
5: end for
6: obrazWy.ujednolicenieEtykiet(tabelaRownowaznosci)
7: tabelaRownowaznosci.wyczysc()
```

2.5 Przygotowanie danych wejściowych dla transformacji wododziałowej

Jak już wspomniano, w transformacji wododziałowej wartości punktów odpowiadają wysokości rzeźby terenu. Granice między zlewiskami powstają w miejscach, w których występują wysokie wartości. Tak więc, aby otrzymać prawidłową segmentację obiektu, jego krawędziom w obrazie wejściowym powinny odpowiadać wysokie, a jego wnętrzu małe wartości. Większość rzeczywistych obrazów nie spełnia tego wymogu. W takim wypadku można je przekształcić do wymaganej postaci za pomocą filtru obliczającego obraz gradientowy. Poniżej przedstawiono przegląd stosowanych metod wyznaczania obrazów gradientowych ze szczególnym uwzględnieniem wykorzystanych w niniejszej pracy.

2.5.1 Obliczanie obrazu gradientowego dla obrazów monochromatycznych

Bardzo popularnym sposobem obliczania obrazu gradientowego jest zastosowanie jednej z istniejących definicji *gradientu morfologicznego*. Mimo różnych właściwości wszystkie gradienty morfologiczne wydobywają lokalną zmienność wartości pikseli w określonym sąsiedztwie [88].

Podstawowym i najbardziej popularnym wariantem wykorzystywanym między innymi w pracach: [10, 12, 14, 65, 88] jest gradient morfologiczny zdefiniowany jako różnica dylacji i erozji obrazu I:

$$g_m(I) = \delta_B(I) - \epsilon_B(I). \tag{2.14}$$

Gradient morfologiczny jest wyznaczany z użyciem elementu strukturującego, którym często jest kwadrat o rozmiarach 3×3 piksele [74]. Kolejnym wariantem jest *gradient wewnętrzny* [88] określony jako różnica oryginalnego obrazu i jego erozji:

$$g_m^-(I) = I - \epsilon_B(I). \tag{2.15}$$

Gradient wewnętrzny wykrywa wewnętrzne krawędzie jasnych obiektów [88]. Podobnie zdefiniowany jest *gradient zewnętrzny*. Jest on różnicą dylacji obrazu i obrazu oryginalnego:

$$g_m^+(I) = \delta_B(I) - I. \tag{2.16}$$

Ten rodzaj gradientu wykrywa z kolei zewnętrzne krawędzie jasnych obiektów. Zarówno gradient wewnętrzny jak i zewnętrzny pozwalają na uzyskanie cieńszych konturów niż podstawowy wariant gradientu morfologicznego dany równaniem 2.14. Istnieją również *grube gradienty morfologiczne* [88]:

$$g_{mB_i}(I) = \delta_{B_i}(I) - \epsilon_{B_i}(I), \tag{2.17}$$

które są zdefiniowane podobnie jak wariant podstawowy, z tą różnicą, że wykorzystywany jest większy element strukturujący (jeżeli B_i jest kwadratowy, to jego rozmiar jest równy $(2i+1) \times (2i+1)$ pikseli przy czym $i > 1$). Gruby gradient morfologiczny umożliwia wykrywanie w obrazie przejść (między jednorodnymi obszarami) o szerokości większej niż wariant podstawowy. Szerokość wykrywanych przejść można regulować poprzez zmianę rozmiaru elementu strukturującego. Wadą grubych gradientów jest to, że nie umożliwiają wykrywania krawędzi regionów o szerokości mniejszej niż $2i$.

Gradienty morfologiczne mogą również służyć do wyznaczania krawędzi skierowanych w określonym kierunku [88]. W tym celu należy zastąpić standardowy element strukturujący odpowiednio skierowanym elementem liniowym. W przypadku obrazów cyfrowych liczba dostępnych kierunków jest ograniczona.

Dzięki temu, że dylację i erozję można uogólnić na obrazy reprezentowane za pomocą grafów, to można również dla takich obrazów obliczyć wszystkie wymienione powyżej rodzaje gradientów [88]. W związku z tym możliwa jest realizacja transformacji wododziałowej na grafach.

Krawędzie występujące w obrazach różnią się szybkością narastania. Skrajnym przypadkiem są krawędzie o charakterze skokowym. Są one skutecznie wykrywane przez filtr obliczający gradient morfologiczny z małym elementem strukturującym. Niestety taki filtr daje na wyjściu niskie wartości dla krawędzi, które są dość szerokie i w przedziale między jednorodnymi obszarami narastają liniowo. Zastosowanie większego elementu strukturującego poprawia wykrywanie tego typu krawędzi, jednak powoduje, że spada rozdzielczość filtru - krawędzie niewielkich obiektów zlewają się [119]. Dlatego D. Wang zaproponował filtr gradientowy, którego odpowiedź jest proporcjonalna do wysokości krawędzi, a nie do jej szerokości. Jest to gradient wieloskalowy dany równaniem [119]:

$$g_{ms}(I) = \frac{1}{n}\sum_{i=1}^{n} \epsilon_{B_{i-1}} \left(\delta_{B_i}(I) - \epsilon_{B_i}(I)\right), \tag{2.18}$$

gdzie $0 \leqslant i \leqslant n$, a element B_i jest kwadratowy i jego rozmiar to $(2i+1) \times (2i+1)$ pikseli. Wartość n pozwala wpływać na to jak filtr reaguje na krawędzie o narastającej jasności. Większa wartość poprawia skuteczność filtru w wykrywaniu tego typu krawędzi jednak pogarsza ją dla stromych i równoległych krawędzi [74]. Z tego względu zalecane jest stosowanie niewielkich wartości parametru n [74].

Drugim popularnym sposobem wyznaczania obrazu gradientowego, zastosowanym w niniejszej pracy jest skorzystanie z definicji gradientu [36, 54], dla funkcji dwóch zmiennych wyrażonej wzorem:

$$\nabla I = \frac{\partial I}{\partial x}\mathbf{i} + \frac{\partial I}{\partial y}\mathbf{j}, \tag{2.19}$$

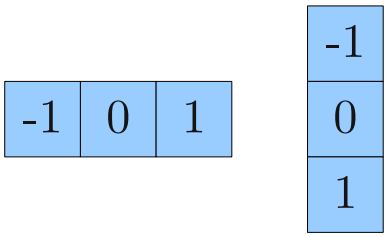
gdzie $\mathbf{i}$ i $\mathbf{j}$ to wektory jednostkowe. Ponieważ obraz gradientowy powinien zawierać wartości skalarne, oblicza się moduł tego gradientu (wzór 2.19) według wzoru:

$$g(I) = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{2.20}$$

Ze względu na to, że obrazy cyfrowe mają charakter dyskretny, nie jest możliwe wyznaczenie pochodnych cząstkowych. Są one zastępowane różnicami centralnymi. Obliczanie ich w praktyce oznacza wyznaczenie, w punktach obrazu, wartości splotu z maskami przedstawionymi na rysunku 2.3 [45].

Rysunek 2.4 przedstawia efekt działania, zastosowanego w niniejszej pracy, filtru gradientowego na przykładzie obrazu z tomografu komputerowego.

Rozwinięcie powyższego sposobu wyznaczania gradientu w obrazie zaprezentował M. Kazanov [50, 51]. Jego rozwiązanie zostało opracowane w celu wykorzystania razem ze zmodyfikowaną wersją transformacji wododziałowej. Umożliwiło



Rysunek 2.3: Maski wykorzystywane przy wyznaczaniu obrazu gradientowego (źródło: [45]).

(a) obraz oryginalny (*rozmiar* 256×256, *bitów na piksel* 12)

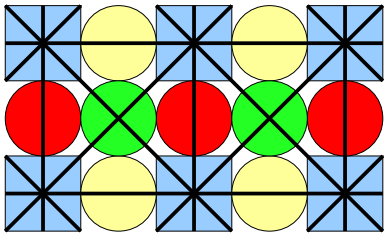
(b) obraz gradientowy

Rysunek 2.4: Przykładowy obraz gradientowy obliczony dla obrazu monochromatycznego pochodzącego z tomografu komputerowego (*obraz CT*: pakiet danych przykładowych dla biblioteki VTK).

ono uzyskanie obrazu gradientowego zawierającego większą liczbę punktów niż obraz oryginalny.

W tej metodzie przybliżoną wartość pochodnej w miejscach odpowiadających pikselom obrazu oryginalnego oblicza się za pomocą różnicy centralnej, czyli za pomocą masek analogicznych do tych przedstawionych na rysunku 2.5. Przybliżoną wartość pochodnej pomiędzy dwoma punktami oblicza się jako różnicę wartości tych punktów. Wada tego rozwiązania jest wyraźnie widoczna przy próbie zastosowania dla obrazu dwuwymiarowego. Rysunek 2.3 ilustruje ograniczenia zaproponowanej metody [50]. Jak widać, w niektórych punktach jest możliwe wyznaczenie pochodnej tylko w jedym lub w dwóch kierunkach. Aby korzystać z tak wyznaczonej informacji o gradiencie, należy zmodyfikować transformację wododziałową [50, 51].

Należy również wspomnieć, że korzystanie z filtru gradientowego nie zawsze jest konieczne, aby przygotować obraz do segmentacji za pomocą transformacji wododziałowej. W [114] zaprezentowano algorytm transformacji wododziałowej, który pracuje bezpośrednio na obrazie oryginalnym.



Rysunek 2.5: Ilustracja ograniczeń filtru gradientowego wysokiej rozdzielczości. Kwadratami oznaczono piksele odpowiadające pikselom obrazu oryginalnego. Kółkami oznaczono piksele, w których nie można wyznaczyć przybliżonej pochodnej we wszystkich kierunkach. Linie wskazują kierunki, w których można wyznaczyć przybliżoną pochodną (źródło: [50]).

2.5.2 Obliczanie obrazu gradientowego dla obrazów barwnych

Wyznaczanie obrazu gradientowego na podstawie obrazu barwnego jest bardziej kłopotliwe. Wynika to z faktu, że jego punkty opisane są za pomocą wektorów, a nie wartości skalarnych.

Pierwszą metodą obliczania obrazu gradientowego jest skorzystanie z heurystyki, która definiuje moduł gradientu w punkcie, jako sumę kwadratów wszystkich pochodnych cząstkowych w tym punkcie, we wszystkich składowych [45]. Dla obrazu dwuwymiarowego o trzech składowych obliczany jest z wykorzystaniem równania o następującej postaci:

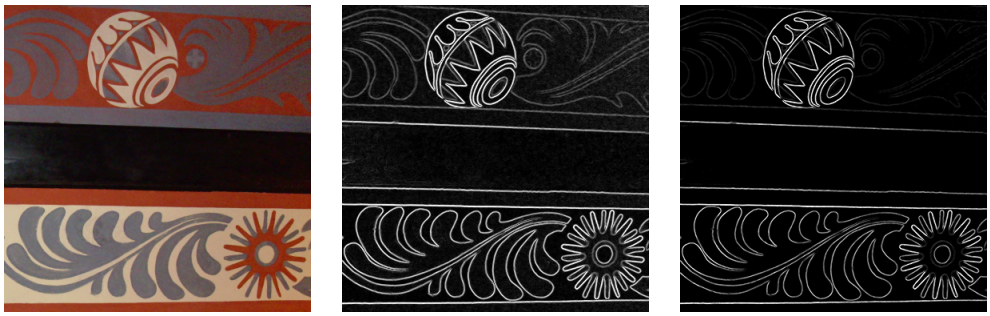
$$g_c(\mathbf{I}) = \sqrt{\sum_{i=1}^{2}\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x_i}\right)^2}, \tag{2.21}$$

gdzie I_j to *j-ta* składowa obrazu. Tak jak w przypadku wzoru 2.20 dla obrazów monochromatycznych, tak i tutaj wartości pochodnych przybliża się za pomocą różnicy centralnej. Warto też zaznaczyć, że powyższy sposób jest równoważny obliczeniu obrazów gradientowych (korzystając ze wzoru 2.21) dla poszczególnych składowych, a następnie połączeniu ich w jeden obraz za pomocą [92]:

$$g_c(\mathbf{I}) = \sqrt{\sum_{j=1}^{3} g(I_j)^2}. \tag{2.22}$$

Powyższy sposób był stosowany w badaniach przedstawionych w dalszej części pracy i będzie określany skrótowo: *gradient-skp*.

Druga z istniejących metod, również zastosowana w niniejszej pracy, opiera się na analizie głównych składowych. To podejście do wykrywania krawędzi w obrazach o wielu składowych zostało zaproponowane przez S. Di Zenzo [126] i rozwinięte przez A. Cummani'ego [29]. Jest to dość popularne rozwiązanie [95, 65, 45]. Autorzy [29, 65] sugerują obliczanie modułu gradientu jako różnicy dwóch



(a) (b) (c)

Rysunek 2.6: Przykładkowe obrazy gradientowe obliczone dla obrazu barwnego: (a) obraz oryginalny, (b) obraz gradientowy obliczony według wzoru 2.21, (c) obraz gradientowy obliczony według wzoru 2.24.

największych wartości własnych macierzy:

$$\begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}, \tag{2.23}$$

zawierającej iloczyny skalarne s_{ij} (równanie 2.26), chociaż równocześnie stwierdzają, że za moduł gradientu można przyjąć inną funkcję wspomnianych wartości własnych macierzy. Dla dwuwymiarowego obrazu moduł gradientu obliczany jest za pomocą wzoru [45]:

$$g_{pca}(\mathbf{I}) = \lambda_+ - \lambda_-, \tag{2.24}$$

w którym wartości własne wyrażone są za pomocą [93]:

$$\lambda_\pm = \frac{s_{11} + s_{22} \pm \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}}{2}, \tag{2.25}$$

a iloczyny skalarne

$$s_{ij} = \frac{\partial \mathbf{I}}{\partial x_i} \cdot \frac{\partial \mathbf{I}}{\partial x_j}. \tag{2.26}$$

Wartości własne λ_+ oraz λ_- mogą być interpretowane, odpowiednio, jako największa i najmniejsza lokalna szybkość zmian obrazu [93] w punkcie, w którym je wyznaczono. Co więcej, istneje również możliwość wyznaczenia (modulo π) wartości θ_+ i θ_-, czyli kierunków największej i najmniejszej zmiany [93]. Wyrażają się one następującymi wzorami:

$$\theta_+ = \frac{1}{2} arctg \frac{2s_{12}}{s_{11} - s_{22}}; \quad \theta_- = \theta_+ + \frac{\pi}{2}. \tag{2.27}$$

Jak już wspomniano, metoda również została zastosowana w badaniach przedstawionych w dalszej części pracy i będzie skrótowo nazywana *gradient-pca*.

Na rysunku 2.6 przedstawiono wyniki otrzymane za pomocą dwóch opisanych metod wyznaczania obrazu gradientowego. Najbardziej widoczną różnicą między

nimi jest to, że gradient-pca wykrywa najbardziej znaczące krawędzie. Mniej istotne są mało wyraźne. Gradient-skp charakteryzuje się większą wrażliwością na mniej znaczące zmiany w obrazie. Zatem gradient-pca jest bardziej użyteczny, gdy segmentacji poddawane są obiekty o wyraźnych krawędziach. Natomiast, Gradient-skp jest bardziej użyteczny, gdy segmentacja ma zawierać mniej wyraźnie zaznaczone elementy obrazu. Zazwyczaj gradient-skp powoduje większą nadsegmentację.

Szereg metod wyznaczania modułu gradientu w przypadku obrazów barwnych zaproponowali J. Angulo i J. Serra [5]. Zaproponowane przez nich sposoby wykorzystują gradient morfologiczny (równanie 2.14), gradient kołowy dla wartości kątowych [40] oraz gradient euklidesowy [96].

Wspomniany gradient kołowy jest obliczany dla obrazów, w których piksele zawierają wartości kątowe. Określony jest wzorem [40]:

$$g_{cir}[I(x)] = \max_{y \in K(x)} d_c[I(x), I(y)] - \min_{y \in K(x)} d_c[I(x), I(y)], \tag{2.28}$$

gdzie odległość d_c między wartościami kątowymi dana jest wzorem [40]:

$$d_c(a,b) = \begin{cases} |a-b| & \text{jeżeli } |a-b| \leqslant \pi \\ 2\pi - |a-b| & \text{jeżeli } |a-b| > \pi. \end{cases} \tag{2.29}$$

Gradient euklidesowy, natomiast, wyraża się wzorem [96]:

$$g_{euk}[\mathbf{I}(x)] = \max_{y \in K(x)} d_e[\mathbf{I}(x), \mathbf{I}(y)] - \min_{y \in K(x)} d_e[\mathbf{I}(x), \mathbf{I}(y)], \tag{2.30}$$

gdzie d_e to odległość euklidesowa między współrzędnymi dwóch wektorów. Ten rodzaj gradientu nadaje się do obrazów, w których piksele opisane są za pomocą wektorów.

Jak wspomniano powyżej zdefiniowano szereg metod obliczania modułu gradientu w obrazach barwnych. Obrazy gradientowe wyznacza się na podstawie obrazu oryginalnego w przestrzeni Lab lub przestrzeni IHLS, która została zaproponowana przez A. Hanburry'ego i J. Serrę [39]. W kolejnych wzorach składowe, które wykorzystywano do obliczeń wskazano za pomocą indeksu dolnego: na przykład I_L oznacza składową L obrazu barwnego $\mathbf{I}$.

Pierwszym zaproponowanym rodzajem gradientu jest *gradient luminancji* [5]:

$$g_{lum}(\mathbf{I}) = g_m(I_L). \tag{2.31}$$

Jak widać jest to gradient morfologiczny obliczony dla składowej L obrazu. Kolejnym rodzajem jest *kołowy gradient odcienia* [5]:

$$g_H(\mathbf{I}) = g_{cir}(I_H), \tag{2.32}$$

który jest gradientem kołowym obliczonym dla składowej H obrazu w przestrzeni IHLS. Trzeci jest *gradient ważony nasyceniem*, wyrażony wzorem [5]:

$$g_S(\mathbf{I}) = I_S \cdot g_{cir}(I_H) + (M_S - I_S) \cdot g_m(I_L), \tag{2.33}$$

gdzie M_S to największa z możliwych wartość składowej S w przestrzeni IHLS. Wartość nasycenia określa udział dwóch składników w ostatecznej wartości gradientu. Czwartą propozycją zawartą w [5] jest *gradient oparty na supremum*:

$$g_{sup}(\mathbf{I}) = max[g_m(I_S), g_m(I_L), g_{cir}(I_H)], \tag{2.34}$$

w którym w każdym punkcie wybierana jest maksymalna wartość gradientu poszczególnych składowych. *Gradient chromatyczny* wyznaczany jest na podstawie składowych a i b przestrzeni Lab. Jest to gradient euklidesowy wyznaczony dla obrazu, $\mathbf{I_{ab}}$, który ma tylko dwie składowe [5]:

$$g_{chr}(\mathbf{I}) = g_{euk}(\mathbf{I_{ab}}). \tag{2.35}$$

Ostatnią propozycją jest *gradient percepcyjny*. Jest to gradient euklidesowy wyznaczony dla obrazu $\mathbf{I_{Lab}}$ [5]:

$$g_p(\mathbf{I}) = g_{euk}(\mathbf{I_{Lab}}). \tag{2.36}$$

Do wyznaczania obrazu przedstawiającego krawędzie obiektów nie zawsze używany jest filtr gradientowy. Można na przykład obliczać lokalną wariancję dla każdego piksela w jego otoczeniu o rozmiarach 7×7 piskeli [115]:

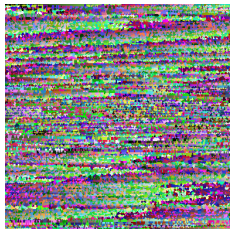
$$\sigma^2(x) = \frac{1}{49} \sum_{y \in N(x)} \|\mathbf{I}(y) - \mu_{\mathbf{N}}(x)\|^2, \tag{2.37}$$

gdzie $\mu_{\mathbf{N}}(x) = \frac{1}{49} \sum_{y \in N(x)} \mathbf{I}(y)$.

W literaturze pojawiają się również rozwiązania dostosowane do konkretnego typu segmentowanych obrazów. Przykładem może być rozwiązanie zaprezentowane w [75], gdzie przedstawiono algorytm segmentacji obrazów przedstawiających próbki skał. Wykorzystano tu fakt, że dla każdej próbki można uzyskać wiele obrazów, z różnymi ustawieniami filtru polaryzacyjnego. Obraz gradientowy wyznaczany jest w następujący sposób: (1) obrazy próbki przekształcane są do przestrzeni Lab, (2) dla składowych a poszczególnych obrazów obliczany jest gradient morfologiczny, (3) otrzymane obrazy gradientowe łączone są w jeden poprzez wybór wartości maksymalnej w każdym punkcie.

2.6 Problem nadsegmentacji

Jak już wspomniano przyczyną nadsegmentacji jest fakt, że transformacja wododziałowa tworzy osobne zlewisko dla każdego minimum lokalnego. Nie ma znaczenia, czy jest ono otoczone wyraźnymi czy niewyraźnymi krawędziami, ani też czy pojawiło się w obrazie gradientowym z powodu zaszumienia obrazu. Sytuację pogarsza dodatkowo czułość filtrów gradientowych na niewielkie zmiany wartości w obrazie.

kwiatek_3		**sufit_2**	
			
(a) obraz oryginalny: (384 × 384)	(b) bez redukcji nadsegmentacji: *12762* zlewisk	(c) obraz oryginalny: (384 × 384)	(d) bez redukcji nadsegmentacji: *20793* zlewisk

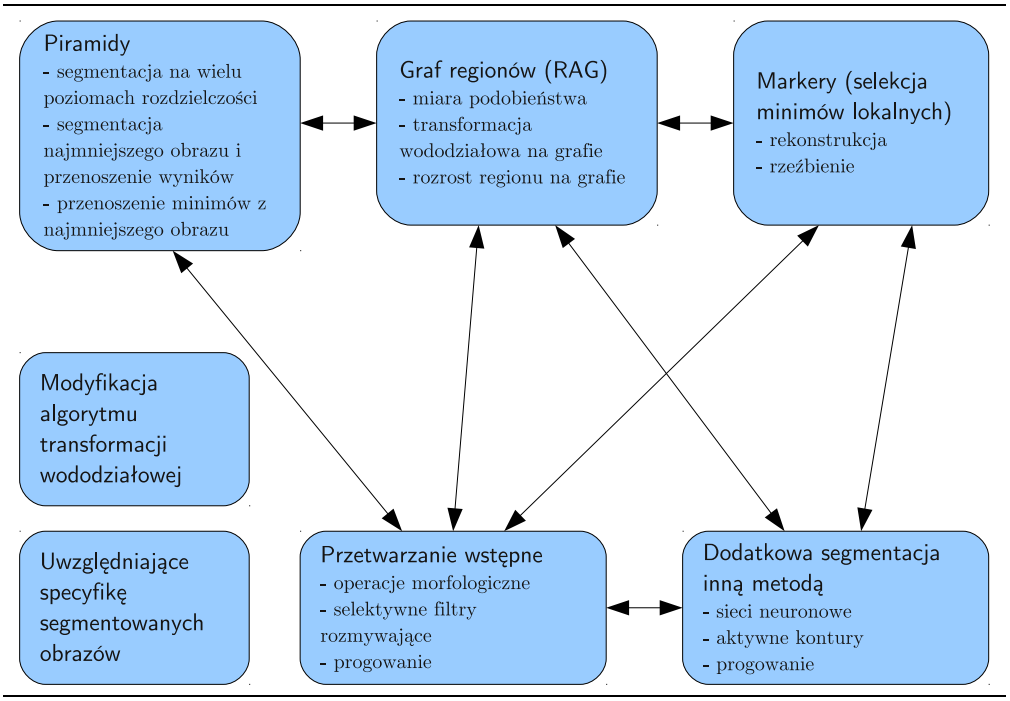
Rysunek 2.7: Wyniki transformacji wododziałowej uzyskane bez stosowania redukcji nadsegmentacji (bez selektywnego filtru rozmywającego, progowania obrazu gradientowego, oraz zalewania płytkich zlewisk). Obraz gradientowy obliczono za pomocą wzoru 2.21. Zlewiskom przypisano barwy przy pomocy funkcji mieszającej.

Rysunek 2.7 przedstawia przykład silnej nadmiernej segmentacji. Nawet małe wartości w obrazie gradientowym powodują, że podczas realizacji transformacji wododziałowej powstają zapory między zlewiskami, i niemożliwe jest ich łączenie. Rysunki 2.7a oraz 2.7c przedstawiają obrazy barwne, które zostały poddane działaniu filtru gradientowego-skp a następnie transformacji wododziałowej. Wyniki znajdują się na rysunkach 2.7b oraz 2.7d. Jak widać oba obrazy zostały nadmiernie posegmentowane. Liczba regionów wynosząca ponad 10000 czy 20000 jest dużo większa niż liczba obiektów występujących w obrazie. W przedstawionych segmentacjach jeden region składał się przeciętnie z kilku do kilkunastu pikseli. W kolejnym punkcie przedstawiono przegląd metod redukcji nadsegmentacji opisanych w literaturze. W dalszej kolejności opisano sposoby wstępnej redukcji nadsegmentacji zastosowane w niniejszej pracy.

2.7 Przegląd metod redukcji nadsegmentacji

Istniejące metody zmniejszania nadsegmentacji, można podzielić na dwie podstawowe kategorie. Pierwszą są metody przetwarzania wstępnego, których celem jest, poprzez odpowiednią modyfikację obrazu oryginalnego lub gradientowego, zmniejszenie stopnia nadsegmentacji. Drugą kategorią są metody, których zadaniem jest odpowiednie łączenie regionów otrzymanych w wyniku transformacji wododziałowej. Jak widać metody kategorii pierwszej są wykonywane przed, a metody kategorii drugiej po przeprowadzeniu transformacji wododziałowej.

Do przetwarzania wstępnego wykorzystuje się kilka różnych typów filtrów. Pierwszym rodzajem są filtry realizujące operacje morfologiczne. Przykładem za-



Rysunek 2.8: Diagram przedstawia typowe metody redukcji nadsegmentacji. Strzałki wskazują rozwiązania, które w wielu pracach są łączone.

stosowania rozwiązania należącego do tej grupy jest algorytm segmentacji nagrań wideo, przedstawiających drogę przed jadącym samochodem [13, 16, 17]. Wykorzystano w nim *filtr DTF* (ang. *dynamic time filter*), dla którego sekwencja klatek stanowi statyczny obraz trójwymiarowy. Obraz poddawany segmentacji jest wyznaczany za pomocą serii wykonywanych naprzemiennie operacji *domknięcia w czasie TC* i *otwarcia w czasie TO* z coraz większym elementem strukturującym:

$$I_{n+1} = TO_{n+1}[TC_{n+1}(I_n)]; \quad I_0 = I \tag{2.38}$$

Domknięcie i otwarcie w czasie zdefiniowano tak samo jak zwykłe operacje domknięcia i otwarcia, czyli za pomocą *dylacji* i *erozji w czasie*, które również działają analogicznie do swoich tradycyjnych odpowiedników, z tą różnicą, że element strukturujący w obrazie trójwymiarowym jest skierowany wzdłuż osi czasu od bieżącej chwili wstecz i ma rozmiar $1 \times 1 \times k$. Efektem działania opisanego filtru jest redukcja szumu, usunięcie z obrazu drogi niewielkich obiektów oraz połączenie linii przerywanych co ułatwia segmentację poszczególnych pasów ruchu. Kolejnym krokiem jest zastosowanie filtru *top-hat*: $I_{DTF} - O_k(I_{DTF})$, który uwydatnia lokalne maksima [13, 16, 17] (I_{DTF} to dwuwymiarowa klatka nagrania po zastosowaniu filtru DTF, O_k to operacja otwarcia o rozmiarze k).

Drugim typem filtrów stosowanym podczas przetwarzania wstępnego są selektywne filtry rozmywające. Istnieje wiele różnych rozwiązań tego typu. W pra-

cy [97] autorzy zaproponowali wersję filtru Gaussa, w którym współczynniki maski, a przez to siła działania filtru w poszczególnych kierunkach, zależą od wartości składowych gradientu w tych kierunkach. Filtr został skonstruowany w taki sposób, aby efekt rozmycia był najsilniejszy w kierunku równoległym do krawędzi a najmniejszy w kierunku do niej prostopadłym. Podobne rozwiązanie zastosowano w [95].

W [43] autorzy zaproponowali filtr rozmywający, którego działanie polega na zakwalifikowaniu poszczególnych pikseli do jednej z dwóch kategorii. Pierwsza kategoria to piksele, których otoczenie jest jednorodne. Autorzy [43] przyjęli, że otoczenie jednorodne charakteryzuje się rozkładem normalnym odcieni szarości. Druga kategoria to piksele, których otoczenie nie jest jednorodne co oznacza, że rozkład odcieni szarości jest mieszaniną dwóch rozkładów Gaussa [43]. W obrazie wynikowym piksele pierwszej kategorii są zastępowane wartością średnią swojego otoczenia, natomiast dla pikseli drugiej kategorii: (1) obliczany jest próg oddzielający dwa składowe rozkłady mieszaniny, (2) na podstawie obliczonego progu przetwarzany piksel przypisywany jest do jednego z rozkładów składowych, (3) w obrazie wynikowym zastępowany jest wartością średnią rozkładu, do którego został przypisany.

W pracach [58, 60] zastosowano *rekursywny filtr oparty na kontraście*. Autorzy określili dwa operatory S_1 i S_2 wykorzystujące dylację i erozję. Są one dane równaniem:

$$S_1(x) = \underset{\theta\in[0,\pi]}{min}\,\delta_{B_\theta}[I(x)]; \quad S_2(x) = \underset{\theta\in[0,\pi]}{max}\,\epsilon_{B_\theta}[I(x)], \tag{2.39}$$

gdzie element strukturujący B_θ jest elementem liniowym (o rozmiarze 1×3 punkty) ustawionym pod kątem 0, 45, 90 lub 135 stopni w zależności od wartości parametru θ. Ponieważ w [58, 60] opisywano segmentację obrazów barwnych należało zdefiniować relację umożliwiającą porównywanie wartości pikseli i w konsekwencji znajdowanie minimów i maksimów. Autorzy przyjęli porządek leksykograficzny. Za pomocą powyższych operatorów (wzór 2.39) zdefiniowano filtr, który przegląda kolejne piksele i analizuje lokalny kontrast. Wartość piksela w obrazie wynikowym określa wzór:

$$I'(x) = \begin{cases} (S_1(x) + S_2(x))/2 & \text{jeżeli } \|S_1(x) - S_2(x)\| < K \\ I(x) & \text{w przeciwnym wypadku.} \end{cases} \tag{2.40}$$

Algorytm wykonuje kilka iteracji. Jako wartość kontrastu K autorzy proponują przyjąć największą wartość odchylenia standardowego obliczoną dla poszczególnych składowych.

Kolejnym rozwiązaniem problemu selektywnego rozmywania są filtry wykorzystujące równania różniczkowe. Dwie najpopularniejsze kategorie oparte są na równaniu dyfuzji lub metodzie poziomic. Ponieważ dwa takie filtry wykorzystywano w badaniach przedstawionych w niniejszej pracy, wspomniane filtry opisano bardziej szczegółowo w punkcie 3.2.1.

Omówione wcześniej kategorie filtrów zazwyczaj wykorzystuje się do wstępnego przetwarzania obrazu oryginalnego. Progowanie, natomiast, jest najczęściej stosowane do obrazu gradientowego. Zazwyczaj jeden próg jest stosowany do całego obrazu. Takie rozwiązanie przyjęto w niniejszej pracy do wstępnej redukcji nadsegmentacji (więcej informacji w punkcie 3.2.2). Niewielkie wartości gradientu (czyli te które znajdują się poniżej progu), są zastępowane wartością 0 [43] lub wartością progu (punkt 3.2.2). Przykładem bardziej zaawansowanego progowania, jest to, zastosowane do rozwiązania problemu segmentacji małych obiektów w sekwencjach wideo [44]. W celu odróżnienia szumu od niewielkich obiektów na podstawie kolejnych klatek obliczane są obrazy różnicowe. Dla wszystkich pikseli obrazu różnicowego ustalana jest liczba niezerowych sąsiadów. Potem przeprowadzane jest progowanie, przy czym próg dobierany jest na podstawie niewielkiego otoczenia przetwarzanego piksela. Szczegółowo proces ten przebiega następująco: (1) tworzona jest kopia wartości pikseli w rozważanym otoczeniu obrazu różnicowego, (2) w kopii zerowane są piksele, które mają więcej niż k niezerowych sąsiadów (w [44] $k = 3$), (3) obliczany jest histogram tak zmodyfikowanego otoczenia (z pominięciem punktów o wartości zero), (4) próg odpowiada miejscu, w którym następuje największy spadek wysokości słupków histogramu.

Kolejnym sposobem przetwarzania obrazu gradientowego, przed wykonaniem transformacji wododziałowej, jest stosowanie markerów. Jest to jedna z najstarszych i najbardziej rozpowszechnionych metod redukcji nadsegmentacji [1, 12, 14, 33, 34, 58, 59, 60, 65, 71, 72, 78, 89, 114, 117, 124]. Ta metoda stosowana jest razem z wariantem algorytmu transformacji wododziałowej symulującym zalewanie rzeźby terenu. Jak już wspomniano w punkcie 2.6, w trakcie zalewania powstaje tyle regionów, ile jest minimów lokalnych. Ideą markerów jest dopuszczenie do powstawania nowych działów wodnych tylko w wybranych miejscach, odpowiadających istotnym minimom obrazu gradientowego. Efekt taki można uzyskać wprowadzając do algorytmu transformacji wododziałowej dodatkowe ograniczenie [36], które pozwala na tworzenie nowego działu wodnego tylko w miejscach określonych przez obraz m zawierający markery, a nie w każdym znalezionym minimum.

Podobny efekt bez wprowadzania dodatkowych ograniczeń do algorytmu realizującego transformację wododziałową można osiągnąć modyfikując obraz gradientowy poprzez wykonanie *rekonstrukcji przez erozję* obrazu $min(g, m)$ z markera m [116], gdzie $g = |\nabla I|$ to obraz gradientowy segmentowanego obrazu, a m to obraz, w którym piksele odpowiadające istotnym minimom obrazu g oznaczono za pomocą wartości $min(g)$ a pozostałe punkty mają wartość $max(g)$. W wyniku rekonstrukcji otrzymuje się wtedy zmodyfikowany obraz gradientowy

$$g' = \rho^*_{min(g,m)}(m), \tag{2.41}$$

w którym istotne minima mają wartość $min(g)$ natomiast wszystkie pozostałe zostają wypełnione.

Rekonstrukcja obrazu f z markera m przez erozję zdefiniowana jest wzorem [36, 74]:

$$\rho_f^*(m) = \epsilon_f^{(k)}(m), \tag{2.42}$$

przy czym k jest takie, że $\epsilon_f^{(k)}(m) = \epsilon_f^{(k+1)}(m)$. Operacja $\epsilon_f^{(k)}(m)$ to erozja geodezyjna o rozmiarze n markera m względem maski f dana równaniem [36, 74]:

$$\epsilon_f^{(n)}(m) = \epsilon_f^{(1)}\left[\epsilon_f^{(n-1)}(m)\right], \tag{2.43}$$

gdzie $\epsilon_f^{(1)}(m)$ to erozja geodezyjna o wielkości 1 markera m względem maski f określona za pomocą [36, 74]:

$$\epsilon_f^{(1)}(m) = (m \ominus b) \vee f, \tag{2.44}$$

przy czym $\vee$ to operator maksimum działający na odpowiadających sobie punktach, a $\ominus$ to operator erozji.

Liczbę minimów lokalnych można również ograniczyć za pomocą operacji *h-minimum* [74] zdefiniowaną równaniem:

$$h - min_h = \rho_f^*(f + h). \tag{2.45}$$

Powoduje ona wypełnienie wszystkich dolin do wysokości h, co oznacza, że znajdujące się blisko siebie minima, rozdzielone wzniesieniami o wysokości mniejszej lub równej h, zostaną zastąpione równiną. Zaletą tego rozwiązania jest to, iż rozwiązuje ono problem wyboru istotnych minimów (istotne są z definicji głębokie minima), jednak zmniejsza kontrolę nad procesem segmentacji [58, 60, 65].

Istnieje jeszcze jedno podejście do modyfikacji obrazu gradientowego - *morfologiczne rzeźbienie* (ang. *morphological carving*). Zostało ono zaproponowane przez P. Soille [110], ponieważ rekonstrukcja przez erozję może tworzyć duże płaskie powierzchnie w obrazie gradientowym co z kolei może powodować problemy z dokładnością segmentacji. Algorytm zaproponowany przez P. Soille działa podobnie do algorytmu realizującego transformację wododziałową przez symulację zanurzania. Po podaniu na wejście obrazu gradientowego i obrazu z zaznaczonymi istotnymi i nieistotnymi minimami, rozpoczyna się proces zalewania rzeźby terenu. Źródłami wody są tylko istotne minima. Podczas zalewania pikseli na kolejnych poziomach, zapisywane są informacje o kierunkach, z których pochodzi woda zalewająca poszczególne piksele. Informacja ta jest zapisywana dla każdego piksela osobno, w dwuwymiarowej tablicy pomocniczej. Dzięki temu, możliwe jest wyznaczenie drogi powrotnej do minimum lokalnego, z którego pochodzi woda zalewająca dany piksel. W momencie gdy woda zaleje minimum, które jest nieistotne, rozpoczyna się "rzeźbienie", przy wykorzystaniu zapisanych wcześniej informacji. W ten sposób powstaje nierosnąca ścieżka (swego rodzaju wąwóz) prowadzący od nieistotnego do najbliższego istotnego minimum.

Innym, niż modyfikacja obrazu gradientowego i równocześnie bardziej skomplikowanym zagadnieniem jest znalezienie takich markerów, które spowodują uzyskanie dobrej segmentacji. Typowym rozwiązaniem jest zastosowanie innej prostszej metody takiej jak progowanie histogramu. Sposób ten wykorzystano w [72] uzyskując markery służące do segmentacji nagrań wideo zawodów sportowych. Jeden marker tworzony był dla zielonego boiska a drugi dla sylwetek sportowców. W pracy [78] za markery przyjęto punkty o barwach odpowiadających kolejnym maksimom histogramu. Uzyskane w ten sposób znaczniki wykorzystano do segmentacji krwinek. Bardzo ciekawe rozwiązanie zastosowano w [59] ponieważ podczas segmentacji zdjęć komórek badanych pod mikroskopem transformację wododziałową wykorzystano dwukrotnie. Najpierw dla każdej z barwnych składowych wyznaczono histogram. Następnie dla każdego histogramu $H(n)$ przeprowadzono morfologiczną rekonstrukcję z $G(n) = H(n) - h$ (lub $G(n) = 0$ gdy $H(n) < h$) gdzie $h = |\sigma(H) - \sqrt{max(H)}|$. Następnie histogramy były odwracane $(max[G(n)] - G(n))$ i dla każdego z nich przeprowadzono transformację wododziałową. Następnie zredukowano nadsegmentację histogramów i uzyskane w ten sposób etykiety umieszczono w odpowiednich składowych barwnych segmentowanego obrazu. Poprzez połączenie etykiet poszczególnych składowych powstały barwne etykiety dla których obliczono histogram. Za markery przyjęto punkty związane z N_c największymi maksimami. Korzystając z tych znaczników przeprowadzono kolejną transformację wododziałową tym razem obrazu dwuwymiarowego.

Zaproponowano również inne sposoby znajdowania istotnych minimów. W pracy [58] do wyboru markerów zastosowano klasyfikatory Bayesa. Natomiast w pracach [1], [33], [124] znaczniki uzyskano dzięki analizie obrazu w wielu rozdzielczościach. Ponieważ jednak budowanie piramid obrazów jest osobną i nowszą metodą usuwania nadsegmentacji to jej opis zostanie przedstawiony w dalszej części punktu.

Nieco nowszą i niezmiernie popularną metodą zmniejszania nadsegmentacji jest stosowanie *grafu regionów* (ang. *region adjacency graph*) [10, 24, 28, 31, 42, 43, 52, 56, 83, 85, 92, 115, 121, 122]. Ponadto, mimo że w [3, 25, 58, 60, 65, 97, 129] autorzy nie używają sformułowania *graf regionów* to zasada działania algorytmu łączącego nadmiernie posegmentowane regiony jest taka sama jak algorytmów pracujących na grafach. Z tego względu zostaną one opisane wspólnie.

Algorytmy pracujące na grafach regionów można podzielić na trzy główne kategorie: (1) łączące działy wodne na podstawie pewnej miary podobieństwa, (2) wykonujące transformację wododziałową na grafie oraz (3) wykorzystujące algorytm rozrostu regionów uogólniony na grafy.

Typowy schemat algorytmów należących do kategorii pierwszej składa się z trzech etapów: (1) znalezienie kandydatów do połączenia, (2) wykonanie połączenia – aktualizacja etykiet i grafu (3) sprawdzenie warunku stopu i ewentualnie powrót do punktu (1). Oczywiście poszczególne rozwiązania różnią się pod względ-

dem sposobu wyboru regionu do łączenia, zastosowanej miary podobieństwa a także warunku zakończenia działania algorytmu.

Pierwsza grupa miar jest zazwyczaj oparta na średnich i rozmiarach porównywanych zlewisk. Czasami dodatkowo uwzględniana jest ich wariancja. W [24] oraz w [97] w pierwszej kolejności łączone są najmniejsze regiony ze swoimi najbardziej podobnymi sąsiadami. Ich podobieństwo jest oceniane za pomocą następującej statystyki:

$$|\mu_i - \mu_j| \leqslant T \cdot \sqrt{\left(\frac{n_i + n_j}{n_i n_j (n_i + n_j - 2)}\right)((n_i - 1)\sigma_i^2 + (n_j - 1)\sigma_j^2)}, \tag{2.46}$$

gdzie: μ_i,μ_j stanowią średnie regionów i oraz j, n_i,n_j liczby punktów w tych regionach a σ_i^2 oraz σ_j^2 ich wariancje. (Powyższa statystyka ta odpowiada ona statystyce testowej o rozkładzie t-Studenta dla małych prób ($n < 30$) służącej do weryfikacji hipotezy o równości średnich [24, 97]). W innym rozwiązaniu posłużono się miarą odległości (różnicy) $\delta_{i,j}$ dwóch regionów i oraz j [31, 43]:

$$\delta_{i,j} = \frac{n_i \cdot n_j}{n_i + n_j}(\mu_i - \mu_j)^2. \tag{2.47}$$

Przy czym w pracy [31] do łączenia dopuszczane są wyłącznie regiony, które nie znajdują się w konflikcie tzn. spełniają warunek:

$$\frac{n_{i,j}^e}{n_{i,j}^b} \leqslant t_e, \tag{2.48}$$

gdzie: $n_{i,j}^e$ to liczba pikseli wspólnej krawędzi regionów i oraz j, $n_{i,j}^b$ to całkowita liczba pikseli krawędziowych tych regionów, a t_e to arbitralnie określony próg. Według autorów [31] fakt pozostawania regionów w konflikcie oznacza, że ich wspólna granica jest istotną granicą między dwoma klasami.

Jeszcze innym sposobem znajdowania regionów do połączenia, jest obliczenie dla każdego z nich średniej μ_i (moment pierwszego rzędu) a także momentu drugiego $m_{i(2)} = \frac{1}{n_i}\sum\sum(R_i(x,y) - \mu)^2$ i trzeciego rzędu $m_{i(3)} = \frac{1}{n_i}\sum\sum(R_i(x,y) - \mu)^3$ (sumowanie odbywa się po wszystkich pikselach wchodzących w skład regionu o numerze i, a $R_i(x,y)$ to wartości poszczególnych pikseli regionu). Momenty są sumowane $mv_i = \mu_i + m_{i(2)} + m_{i(3)}$ i na podstawie tych sum określana jest odległość (różnica) sąsiadujących regionów. Wyraża się ona wzorem [52]:

$$\delta_{i,j} = (mv_i - mv_j)^2. \tag{2.49}$$

Dodatkowy problem pojawia się w trakcie segmentacji obrazów barwnych, ponieważ punkty opisane są za pomocą wektorów. W [92] przedstawiono sposób redukcji nadsegmentacji w obrazach barwnych. Wektory charakteryzujące poszczególne zlewiska zawierały ich uśrednione barwy. Średnie obliczano za pomocą składowych RGB. W celu porównania dwóch regionów obliczano odległość euklidesową między ich średnimi barwami.

Inną miarą różnicy, którą można zastosować w segmentacji obrazów barwnych jest odległość Fishera [60, 94, 115]. Tak jak poprzednie, również ta miara jest obliczana na podstawie średniej, rozmiaru i wariancji zlewiska. Dla dwóch regionów i i j jest ona dana wzorem [115]:

$$d_{ij}^2 = \frac{(n_i + n_j)\|\mu_i - \mu_j\|^2}{n_i\sigma_i^2 + n_j\sigma_j^2} \tag{2.50}$$

lub

$$d_{ij}^2 = \|\mu_i - \mu_j\|^2 \text{ gdy } \sigma_i^2 = 0 \text{ i } \sigma_j^2 = 0, \tag{2.51}$$

gdzie n_i, n_j to liczba punktów, które składają się odpowiednio na region i oraz j, μ_i, μ_j to wektory zawierające uśrednione barwy odpowiednich zlewisk, a wariancja zlewiska w obrazie barwnym została określona następująco [60]:

$$\sigma_i^2 = \frac{\sum_{i=1}^{c} \sigma_{s,i}^2}{c}, \tag{2.52}$$

przy czym c to liczba składowych obrazu barwnego, a $\sigma_{s,i}$ to wariancja *s-tej* składowej regionu o numerze i.

Druga grupa miar służących do porównywania zlewisk jest oparta na *dynamice konturów* (ang. *dynamics of contours*). Została zaproponowana przez M. Grimaud [38]. Jest to dość popularny [42, 69, 83] sposób, pozwalający ocenić, jak wyraźna krawędź oddziela dwa regiony. Jeżeli przyjąć, że P_{ij} to zbiór punktów o najniższej wartości, leżących na granicy między regionami i i j, to dynamika dc_{ij} konturu między tymi regionami jest wyrażona wzorem:

$$dc_{ij} = \min_{k\in[1,n]} \max_{x\in B_k} [hw_{ij} - g(x)], \tag{2.53}$$

gdzie g to rozważany obraz (gradientowy), a B_k to zbiór punktów stanowiących spójny obszar, do których prowadzi nierosnąca ścieżka z punktu należącego do zbioru P_{ij}, a hw_{ij} to wysokość konturu między regionami i i j, czyli wartość najniższego punktu leżącego na granicy między zlewiskami i i j.

Zamiast korzystać z równania 2.53, można obliczyć widoczną dynamikę konturu za pomocą [69]:

$$dc_{ij} = hw_{ij} - max(min_i, min_j), \tag{2.54}$$

gdzie: min_i,min_j - to wartości minimów występujących odpowiednio w regionach i i j.

Dynamika zdefiniowana za pomocą równania 2.54 lub 2.53 ma jednak pewną wadę. Wynika ona z faktu, że wysokość krawędzi jest określona jako minimum wartości punktów leżących na krawędzi. Wtedy, jeśli na przykład szum spowoduje, że wartość tylko jednego piksela krawędziowego będzie wyraźnie mniejsza od pozostałych, to obliczona dynamika będzie mała. mimo że dotyczy wyraźnej

krawędzi. Problem ten rozwiązano, proponując zmianę sposobu obliczania wysokości krawędzi [42]. Zaproponowano też dwa nowe sposoby obliczania dynamiki. Pierwszy sposób można wyrazić wzorem:

$$dc_{ij} = hw_{ij}^{r} - max(min_i, min_j), \tag{2.55}$$

gdzie hw_{ij}^{r} to średnia wartość wszystkich punktów leżących na granicy między regionami i i j. Natomiast drugi:

$$dc_{ij} = hw_{ij}^{med} - max(min_i, min_j), \tag{2.56}$$

gdzie hw_{ij}^{med} to wartość środkowa wszystkich punktów leżących na granicy między regionami i i j.

Wykorzystanie dynamiki do łączenia zlewisk jest podobne do wykorzystania innych miar. W kolejnych iteracjach algorytmu wybierane są kontury o najmniejszej dynamice i odpowiadające im regiony są łączone [42, 83]. Dane o dynamice konturów są uaktualniane, ponieważ minimum nowo powstałego regionu to mniejsze z minimów regionów łączonych. Powstaje też nowa lista krawędzi regionu – zawiera ona wszystkie krawędzie regionów łączonych – z wyjątkiem tych, które stały się krawędziami wewnętrznymi. W [42] autorzy zwracają uwagę, że przechowywanie w wierzchołkach grafu informacji o minimach regionów a w krawędziach informacji o wysokości krawędzi, liczbie punktów krawędziowych, sumie tych punktów oraz ich histogramu umożliwia szybkie obliczanie zaproponowanej przez nich dynamiki.

W kilku pracach zastosowano rozwiązanie podobne do dynamiki konturów. W [85] zlewiska są łączone gdy głębokość każdego z nich nie przekracza zadanego progu. Głębokość zlewiska jest określona jako różnica wysokości wspólnej krawędzi zlewisk i wartości minimum danego zlewiska. W pierwszej kolejności algorytm próbuje łączyć z sąsiadami te regiony, które powstały najwcześniej w trakcie działania algorytmu realizującego transformację wododziałową przez zalewanie. W pracach [25, 129] przedstawiono kolejne rozwiązanie podobne do dynamiki konturów. Dla każdego zlewiska jest obliczana głębokość (w sposób taki sam jak w [85]) względem jego najniższej krawędzi. Krawędź ta określa równocześnie, przez którego sąsiada może zostać zalane rozważane zlewisko. Następnie zlewiska są sortowane rosnąco względem wysokości. Łączenie regionów w nadmiernie posegmentowanym obrazie odbywa się właśnie w tej kolejności. Sposób ten został wykorzystany w niniejszej pracy do wstępnej redukcji nadsegmentacji i został szczegółowo opisany w punkcie 3.2.3.

Trzecia grupa miar, która pojawia się w literaturze to miary oparte na wyniku segmentacji inną metodą. Do innych metod segmentacji można zaliczyć klasyfikację pikseli z wykorzystaniem sieci neuronowych [53, 56, 66], w szczególności sieci neuronowych Hopfielda [56, 66]. W pracy [56] skorzystano z faktu, że w segmentowanych obrazach znana była liczba klas, którą należy uzyskać w koń-

cowej segmentacji (segmentowane były skany głowy wykonane rezonansem magnetycznym). Autorzy utworzyli sieć neuronową z liczbą neuronów dopasowaną do docelowej liczby klas, wyszkolili sieć, a następnie z jej pomocą przeprowadzali klasteryzację pikseli obrazu MR. W dalszej kolejności przeprowadzana była transformacja wododziałowa tego samego obrazu MR. Potem na podstawie wyników dwóch segmentacji, dla każdego regionu i otrzymanego w wyniku transformacji wododziałowej obliczany był wektor, którego składowe oznaczają stopień przynależności tego regionu do poszczególnych klas otrzymanych za pomocą sieci neuronowej. Wspomniany wektor miał postać:

$$\mathbf{U}_i = (p_{i1}, p_{i2}, \ldots, p_{ic}), \tag{2.57}$$

gdzie c to docelowa liczba klas (i równocześnie liczba klas uzyskana za pomocą sieci neuronowej), a stopień przynależności to iloraz liczby pikseli $n_{i,k}$ należących równocześnie do zlewiska i i danej klasy k przez liczbę punktów zlewiska n_i. Wyraża się on wzorem:

$$p_{ik} = \frac{n_{i,k}}{n_i}. \tag{2.58}$$

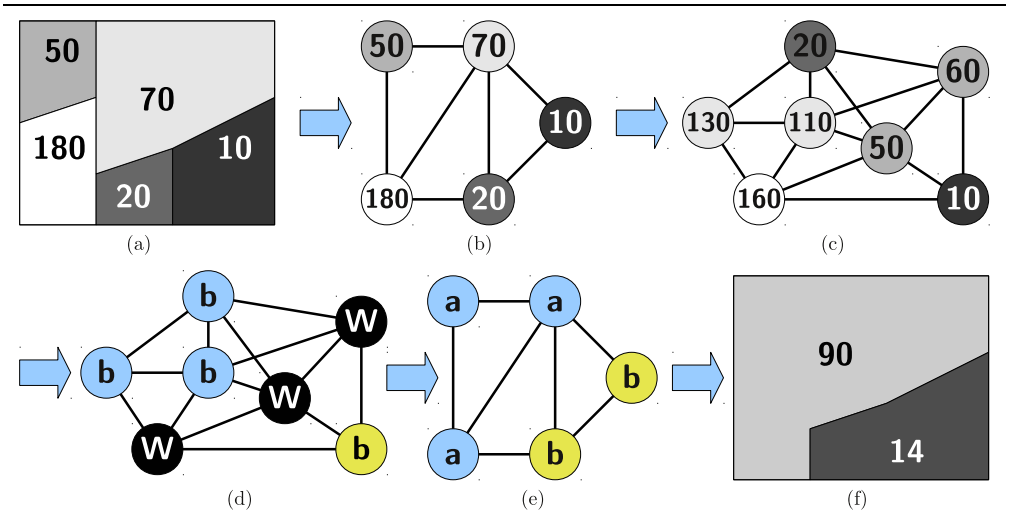
Zaproponowana miara odległości (różnicy) dwóch zlewisk i i j była obliczana na podstawie wektorów U_i i U_j za pomocą wzoru [56]:

$$\delta_{ij} = \sum_{k=1}^{c} |p_{ik} - p_{jk}|. \tag{2.59}$$

W większości prac regiony łączone są w ściśle określonym porządku, na przykład łączenie zaczyna się od najbardziej podobnych regionów [42, 43, 92], albo też w pierwszej kolejności ze swoimi sąsiadami łączone są najmniejsze [24] lub najgłębsze [85] zlewiska. Autorzy [56] zastosowali dość nietypową strategię scalania regionów. Na początku próg δ ustawiany jest na pewną wartość (na przykład 0,01). Wybierany jest dowolny region i i jeżeli jest on dostatecznie podobny do jednego (dowolnego) ze swoich sąsiadów j ($\delta_{ij} \leqslant \delta$), to jest on łączony w nowy region i (następuje uaktualnienie grafu i wektora U_i). Do regionu i dołączani są kolejni jego sąsiedzi. Jeżeli region i nie ma już sąsiadów, których można z nim połączyć. Znajdowany jest inny region, do którego dołączani są kolejni sąsiedzi. Jeżeli nie można znaleźć żadnych kandydatów do połączenia zwiększana jest wartość progu δ (przykładowo o 0,01), a następnie wykonywane są dalsze połączenia aż do osiągnięcia zadanej liczby klas c.

Ostatnim, ale równie ważnym, co miara zastosowana do oceny podobieństwa czy porządek łączenia regionów zagadnieniem, jest warunek stopu, którego spełnienie powoduje zakończenie algorytmu łączenia zlewisk. Istnieją tutaj dwa typowe sposoby postępowania.

Pierwszym z nich jest wykonanie wszystkich możliwych połączeń do momentu spełnienia pewnego warunku, który; co ważne; nie gwarantuje uzyskania określonej liczby klas. Otrzymywana jest optymalna, według pewnego kryterium, liczba



Rysunek 2.9: Ilustracja idei transformacji wododziałowej na grafach (źródło: [121]), (a) obraz mozaikowy, (b) graf regionów (odpowiednik segmentowanego obrazu), (c) graf krawędzi (odpowiednik obrazu gradientowego), (d) wynik transformacji wododziałowej na grafie krawędzi, (e) etykiety przeniesione na graf regionów, (f) nowy obraz mozaikowy (następny poziom hierarchii).

klas. I tak w [31] regiony są łączone do momentu, w którym wszystkie pozostałe regiony są ze sobą wzajemnie w konflikcie (warunek 2.48), w [85] łączone są wszystkie pary zlewisk, dla których ich głębokości względem wspólnej krawędzi są mniejsze niż określony arbitralnie próg [85], w [97] łączona jest połowa regionów uzyskanych w wyniku transformacji wododziałowej. Z kolei w [60] warunek stopu to przekroczenie przez miarę odległości zlewisk kandydujących po połączenia wartości $\frac{K}{2}$, gdzie K to jeden z parametrów filtru wygładzającego. Również w [25, 43, 129] regiony są łączone do momentu, w którym miara odległości przekracza pewien określony wcześniej próg.

Drugim sposobem jest łączenie zlewisk do momentu osiągnięcia zadanej liczby klas. Takie rozwiązanie jest użyteczne, gdy z góry wiadomo, jaka jest liczba klas w segmentowanym obrazie. Z tego względu zostało ono wykorzystane w algorytmie segmentacji obrazów mózgu pochodzących z rezonansu magnetycznego w pracy [56].

Jak już wcześniej wspomniano drugą kategorią algorytmów, które do redukcji nadsegmentacji wykorzystują grafy, są algorytmy tworzące kilka poziomów segmentacji poprzez wykonywanie transformacji wododziałowej na grafach. Rozwiązanie zaproponował S. Beucher w roku 1992 [10]. Pojawiło się ono także w pracach [11, 115, 121, 122]. Schemat tworzenia hierarchii segmentacji jest następujący:

1. Przygotowanie obrazu (filtr rozmywający, filtr gradientowy) i wykonanie

zwykłej transformacji wododziałowej.

2. Obliczenie obrazu mozaikowego. Jest to obraz przedstawiający wynik segmentacji, ale zamiast arbitralnych etykiet regionom przyporządkowane są średnie [121] lub minimalne [10] wartości pikseli w poszczególnych zlewiskach. W przypadku obrazów barwnych może to być średnia barwa zlewiska [115].

3. Utworzenie *grafu regionów*, w którym wierzchołki odpowiadają poszczególnym zlewiskom i zawierają wartość średniej lub minimum danego regionu.

4. Utworzenie *grafu krawędzi* na podstawie grafu regionów. Graf krawędzi jest odpowiednikiem obrazu gradientowego. Każdy wierzchołek tego grafu reprezentuje wspólną granicę dwóch zlewisk (lub krawędź w grafie regionów). Różnica między wartościami (średnimi lub minimami) tych regionów jest zapisana w poszczególnych wierzchołkach. Krawędzie tego grafu są tworzone w sposób następujący: wierzchołek odpowiadający granicy między dwoma zlewiskami jest łączony z wierzchołkami odpowiadającymi wszystkim innym krawędziom tych dwóch zlewisk. Przykładowo wierzchołek o wartości 10 z rysunku 2.9c odpowiada krawędzi między regionami o średnich 10 i 20 z rysunku 2.9a. Jest połączony z wierzchołkami 160, 50 i 60 z rysunku 2.9c, ponieważ odpowiadają one wszystkim pozostałym krawędziom regionów o średnich 10 i 20. W przypadku segmentacji obrazów barwnych do obliczania różnic między średnimi barwami par regionów można wykorzystać odległość Fishera [115] (równania 2.50 i 2.51).

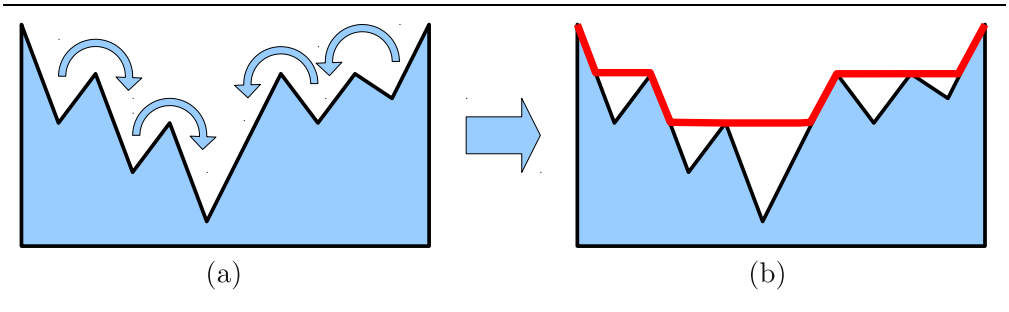
5. Wykonanie transformacji wododziałowej na grafie krawędzi (algorytm jest analogiczny do algorytmu działającego na pikselach)

6. Przeniesienie etykiet z grafu krawędzi na graf regionów.

7. Połączenie regionów o tej samej etykiecie i wyznaczenie nowych średnich/minimów.

8. Sprawdzenie warunku stopu. Jeżeli nie jest spełniony, powrót do punktu 3.

Zazwyczaj autorzy nie precyzują warunku stopu lub proponują wykonanie jednej transformacji wododziałowej na grafie [10, 115]. Najciekawszą propozycję przedstawiono w [122] gdzie prezentowane są segmentacje obrazów medycznych pochodzących z tomografu komputerowego. Autorzy tej pracy proponują wygenerowanie wszystkich możliwych poziomów hierarchii, a następnie wybranie segmentacji poszczególnych obiektów z tych poziomów, na których zostały prawidłowo wysegmentowane. Prawidłowość segmentacji konkretnego obiektu ocenia się na podstawie przygotowanego wcześniej opisu. Wykonanie wszystkich możliwych kroków



Rysunek 2.10: Ilustracja idei algorytmu wodospadów (na podstawie: [11]), (a) jednowymiarowy obraz gradientowy (strzałki pokazują sposób przelewania się wody między zlewiskami), (b) zmodyfikowany obraz gradientowy (czerwona linia), który nie zawiera najmniej istotnych minimów lokalnych

algorytmu oznacza powtarzanie transformacji wododziałowej na kolejnych grafach regionów do momentu, gdy wszystkie obiekty znajdą się w jednej klasie.

Algorytmem podobnym do transformacji wododziałowej na grafach jest *algorytm wodospadów* (ang. *waterfall alhorithm*). Został on, tak jak transformacja wododziałowa na grafach, zaproponowany przez S.Beuchera [11], ale był też stosowany przez J. Angulo i J. Serrę w [5]. W tym podejściu obraz gradientowy jest modyfikowany w taki sposób, aby wyeliminować najmniej istotne minima. W tym celu do pojęć związanych z transformacją wododziałową wprowadzono pojęcie wodospadów. Idea tego algorytmu została przedstawiona na rysunku 2.10a. Gdyby przedstawioną na nim powierzchnię z góry zalewała woda, to po pewnym czasie zaczęłaby się ona przelewać między zlewiskami, tak jak wskazują strzałki. W pewnym momencie wszystkie zlewiska zostaną wypełnione i rzeźba terenu (razem z wypełniającą zlewiska wodą) przyjmie kształt zaznaczony na rysunku 2.10b czerwoną linią. Jak widać, na przykładowym rysunku zostało tylko jedno minimum lokalne.

Formalnie sposób uzyskania zmodyfikowanego obrazu gradientowego g' można zapisać za pomocą wzoru [11]:

$$g' = \rho_g^*(m). \tag{2.60}$$

Oznacza to, że wystarczy przeprowadzić rekonstrukcję przez erozję obrazu gradientowego g z obrazu m. Obraz m określony jest wzorem:

$$m(x) = \begin{cases} g(x) & \text{jeżeli } x \in W(g) \\ M & \text{w przeciwnym wypadku,} \end{cases} \tag{2.61}$$

gdzie $W(g)$ to zbiór punktów tworzących linie wododziałowe w transformacji wododziałowej obrazu g, natomiast M to maksymalna wartość występująca w obrazie $g = |\nabla I|$.

Rekonstrukcję obrazu gradientowego można powtarzać wielokrotnie, uzyskując w ten sposób hierarchię obrazów gradientowych oraz segmentacji $g_2 = \rho^*_{g_1}(m_1)$, $g_3 = \rho^*_{g_2}(m_2)$, ..., $g_n = \rho^*_{g_{n-1}}(m_{n-1})$. Na każdym poziomie, z wykorzystaniem bieżącego obrazu gradientowego oraz jego transformacji wododziałowej, są eliminowane kolejne minima lokalne i zmniejsza się liczba klas. Gdy g_1 jest obrazem gradientowym obrazu mozaikowego to algorytm wodospadów utworzy hierarchię identyczną do tej, którą otrzymano by za pomocą algorytmu transformacji wododziałowej wykonanej na grafie krawędzi tego samego obrazu mozaikowego [11]. Stosując przedstawiony algorytm do obrazów barwnych, po wykonaniu *n-tej* transformacji wododziałowej obliczano obraz mozaikowy μ_n, zawierający średnie barwy poszczególnych zlewisk, a na jego podstawie wyznaczano kolejny obraz gradientowy $g_{n+1} = |\nabla \mu_n|$. Średnie barw obliczano korzystając ze składowych RGB [5].

Trzecią ważną kategorią algorytmów, które redukują nadsegmentację wykorzystując do tego celu grafy, są algorytmy rozrostu regionu [3, 100, 101, 102, 105, 115]. W [115] przedstawiono dwa rozwiązania. Pierwsze z nich wykorzystuje do tego celu graf regionów, w którym wierzchołki odpowiadają poszczególnym regionom. Do oceny odległości (różnicy) regionów zastosowana została odległość Fishera d^2 (równania 2.50 i 2.51). Zaproponowano następujące kryteria [22, 115]:

$$d^2(R_i, R_j) \leqslant t_1;\ d^2(R_i, V) \leqslant t_2;\ d^2(R_i, S) \leqslant t_3 \tag{2.62}$$

służące rozstrzygnięciu czy regiony R_i i R_j należy połączyć. Przy czym V to zbiór regionów, które są sąsiadami regionu R_i, S to zbiór regionów (klasa), do której należy region R_j i potencjalnie może zostać włączony R_i, t_1, t_2, t_3 to arbitralnie wybrane progi. Do łączenia zlewisk zaproponowano następujące zasady [115]:

1. Jeżeli zlewisko jest dostatecznie blisko jakiejś klasy, to jest do niej dołączane.

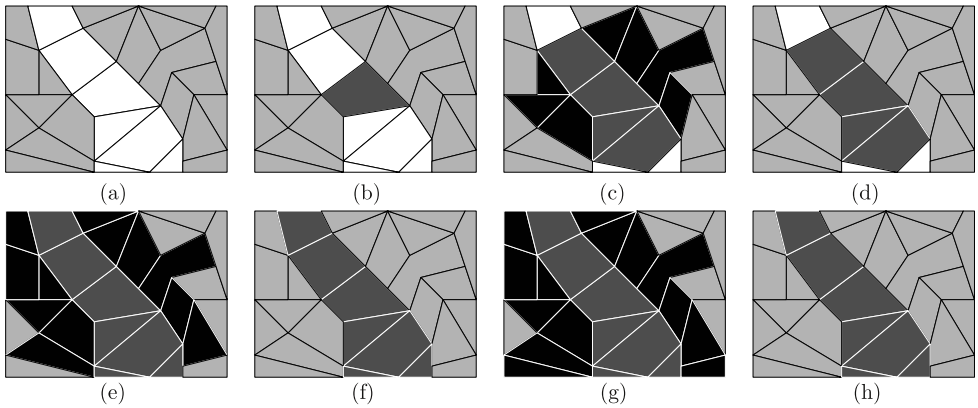
2. W wypadku, gdy region jest dostatecznie blisko kilku sąsiadujących z nim klas, to jest łączony z najbliższą klasą.

3. Jeżeli nie ma w sąsiedztwie regionu dostatecznie podobnej klasy, to tworzona jest nowa.

Drugie rozwiązanie zaproponowane w pracy [115] wykorzystuje graf krawędzi. Poszczególne wierzchołki v_{ij} odpowiadają parom regionów R_i,R_j (czyli krawędziom grafu regionów) i z każdym z nich związane są dwa atrybuty – odległość Fishera $d^2(R_i, R_j)$ oraz waga w_{ij} dana wzorem:

$$w_{ij} = \frac{min(n_i; n_j)}{n_i + n_j}, \tag{2.63}$$

gdzie n_i,n_j to rozmiary regionów.



Rysunek 2.11: Sposób działania algorytmu rozrostu regionu na grafie (a) segmentowany obraz (jasny obiekt na ciemnym tle), (b) region startowy, (c) dylacja regionu startowego (iteracja 1), (d) efekt zwężania histogramu (iteracja 1), (e) dylacja regionu (iteracja 2), (f) zwężanie histogramu (iteracja 2), (f) dylacja regionu (iteracja 3), (g) zwężanie histogramu (iteracja 3)

Etykiety nadawane wierzchołkom v_{ij} określają, które pary regionów należą do tej samej klasy, a które leżą przy granicy klas. W trakcie rozrostu regionów kolejne wierzchołki są łączone w grupy. Para wierzchołków v_{ij} oraz v_{jk} jest grupowana gdy $d^2(R_j, R_k)$ mniej różni się od $d^2(R_i, R_j)$ niż od jakiejkolwiek innej odległości $d^2(R_j, R_j)$. Przy czym $d^2(R_i, R_j)$ jest najmniejszą z nich [115]. Algorytm rozrostu regionu rozpoczyna pracę od punktów startowych, które charakteryzują się tym w swoim otoczeniu, że $d^2(R_i, R_j)$ jest minimalne a w_{ij} jest maksymalne [115].

Również autor niniejszej pracy przedstawił metodę, która wykorzystuje algorytm rozrostu regionu na grafie zlewisk [101]. W rozwiązaniu zastosowano algorytm rozrostu regionu o minimalnej wariancji zaproponowany przez Ch. Revol po raz pierwszy w 1997 roku [86]. Użyto uproszczonej wersji tego algorytmu [87] z tą różnicą, że uogólniono go na grafy. Poniżej została przedstawiona zasada jego działania.

Praca algorytmu rozrostu regionu na grafie zlewisk, podobnie jak innych algorytmów rozrostu regionu, rozpoczyna się od regionu startowego, który w tym przypadku musi zawierać przynajmniej jedno zlewisko. Nie musi być on spójny, gdy zawiera więcej zlewisk. W kolejnych iteracjach aktualny region zostaje poddany dylacji, a następnie *k-lewostronnemu-zwężeniu* histogramu. Dylacja regionu polega na dodaniu do niego wszystkich sąsiadujących z nim zlewisk [101], natomiast *1-lewostronne-zwężenie* histogramu polega na usunięciu z regionu zlewisk, które mają średnią równą najniższej średniej zlewisk należących do przetwarzanego regionu [101]. Pojedyncze zwężenia histogramu są wykonywane do momentu spełnienia kryterium jednorodności regionu. Jest on uważany za jednorodny, gdy jego odchylenie standardowe σ_R nie przekracza zadanego progu σ_t. Zwężanie hi-

Algorytm 2.6 Algorytm rozrostu regionu pracujący na grafie zlewisk

1: $R_0 \leftarrow pustyRegion$
2: $R_1 \leftarrow regionPoczatkowy$
3: $n \leftarrow 1$
4: **while** $R_n \neq R_{n-1}$ **do**
5: $R_{n+1} \leftarrow zlewiskowaDylacja(R_n)$
6: **while** $\sigma_{R_{n+1}} > \sigma_t$ **do**
7: $najnizszaSrednia \leftarrow znajdzNajnizszaSrednia(zlewiska, R_{n+1})$
8: $usunZlewiskaOSredniej(zlewiska, R_{n+1}, najnizszaSrednia)$
9: **end while**
10: $n \leftarrow n + 1$
11: **end while**

stogramu jest wykonywane tylko w razie potrzeby i w niektórych iteracjach (zazwyczaj na początku rozrostu regionu) może okazać się niepotrzebne. Warunkiem stopu algorytmu, sprawdzanym na początku każdej iteracji, jest dwukrotne uzyskanie tej samej segmentacji: $R_{n-1} = R_n$. Pełny przebieg segmentacji pokazano za pomocą prostego przykładu na rysunku 2.11. Przedstawiony algorytm służy do segmentacji obrazów, których histogram jest bimodalny, przy czym wysegmentowany region będzie jaśniejszy niż tło. Bardzo łatwo jest odwrócić działanie algorytmu, tak żeby wysegmentowany obiekt był ciemniejszy od tła – wystarczy zastąpić lewostronne zwężenia histogramu prawostronnymi. Jedyną różnicą w ich działaniu jest to, że usuwają z regionu zlewiska o najwyższej, a nie najniższej średniej.

W implementacji powyższego algorytmu głównym problemem jest wydajność wyznaczania dylacji bieżącego regionu a także wyznaczenie liczby zwężeń histogramu, które należy wykonać. W [100] oraz [105] zaprezentowano sposób poprawy wydajności algorytmu. Poprawę szybkości obliczania dylacji osiągnięto poprzez (1) odpowiednie sumowanie zbiorów sąsiadów zlewisk wchodzących w skład regionu oraz (2) wprowadzenie śledzenia brzegu regionu. Wyznaczenie dylacji regionu wymaga obliczenia sumy list sąsiadów zlewisk. Dla zastosowanego sposobu przechowywania list sąsiadów (uporządkowane zbiory) szybkim sposobem obliczania sumy jest dodawanie zbiorów o równym rozmiarze, co można w przybliżeniu osiągnąć poprzez sumowanie list sąsiadów zlewisk parami. Dodatkowe usprawnienie wynika z faktu, iż aby wyznaczyć dylację regionu, nie trzeba sumować list sąsiadów wszystkich składowych zlewisk. Wystarczy zsumować zlewiska wchodzące w skład regionu oraz dylację jego brzegu. Jak już wspomniano w zmodyfikowanym algorytmie [105] wprowadzono śledzenie modyfikacji brzegu regionu z iteracji na iterację. Poprawę wydajności wykonywania zwężeń histogramu osiągnięto poprzez przechowywanie w grafie charakterystyk zlewisk (liczby punktów, sumy wartości punktów, sumy kwadratów wartości punktów, średnie wartości punktów w zlewisku), które umożliwiają szybkie wyznaczanie odchylenia standardowego regionu

po zwężeniu histogramu. Wprowadzone usprawnienia pozwoliły na uzyskanie od 2 do 30 krotnych przyspieszeń w zależności od obrazu testowego.

W [102] do powyższego algorytmu zastosowano funkcje oceny segmentacji z [87], które pozwoliły na automatyczny dobór progu jednorodności σ_t. Należy również zaznaczyć, że możliwa jest dalsza rozbudowa algorytmu, która pozwalałaby na segmentację obrazów zawierających wiele klas. Rozszerzenie byłoby analogiczne do sposobu przedstawionego przez autora niniejszej pracy w [98, 99], który polegał na przeprowadzeniu jednej segmentacji i wydzieleniu klasy o największej jasności, usunięciu jej z obrazu (poprzez zastosowanie odpowiedniej maski) i przeprowadzeniu kolejnej segmentacji. Powtarzanie tego schematu pozwala na wysegmentowanie zadanej liczby klas z obrazu.

Po zastosowaniu markerów oraz grafów regionów, trzecią najpopularniejszą kategorię metod zmniejszania nadsegmentacji stanowi *analiza wielorozdzielcza*. Polega ona na tworzeniu piramidy obrazów, w której na kolejnych poziomach znajduje się segmentowany obraz w coraz mniejszej rozdzielczości [1, 33, 48, 49, 52, 57, 61, 95, 124]. Typowy przebieg segmentacji jest następujący: (1) zbudowanie piramidy obrazów, (2) wykonanie segmentacji obrazu w najniższej rozdzielczości, obrazu w najwyższej rozdzielczości lub wszystkich obrazów wchodzących w skład piramidy, (3) ustalenie powiązań pomiędzy poziomami i wygenerowanie segmentacji największego obrazu. Najbardziej istotne różnice istnieją w zakresie przenoszenia wyników do najwyższej rozdzielczości, ale występują też w przypadku sposobów budowy piramidy.

W pracach związanych z transformacją wododziałową wśród sposobów budowy piramidy obrazów różnej rozdzielczości można wyróżnić trzy podstawowe rozwiązania:

1. Punkty obrazu o niższej rozdzielczości obliczane są z wykorzystaniem filtru dolnoprzepustowego np. filtru Gaussa lub podobnego do filtru Gaussa [33].

2. Obraz jest filtrowany z zastosowaniem morfologicznego otwierania i zamykania, a następnie skalowany, tak aby uzyskać następny stopień piramidy [124].

3. Kolejne stopnie piramidy uzyskuje się poprzez wykonywanie kolejnych kroków transformaty falkowej. Szczególnie popularna jest transformata Haara [48, 49, 52, 61].

W przypadku wszystkich wymienionych metod, kolejne obrazy mają szerokość i wysokość dwukrotnie mniejszą niż poprzedni.

Pierwszą strategią łączenia poziomów piramidy jest znalezienie związku między zlewiskami otrzymanymi dla obrazu najwyższej rozdzielczości a minimami lokalnymi obrazu gradientowego wyznaczonego dla najmniejszego obrazu w hierarchii. W [33] spośród obrazów tworzących piramidę wybierane są dwa poziomy:

poziom inicjujący s i poziom segmentacji k. Analiza wielorozdzielcza umożliwia znalezienie znaczących minimów lokalnych w obrazie o wyższej rozdzielczości (poziom k), co z kolei zapewnia zmniejszenie nadsegmentacji. Autorzy zaproponowali dwa różne sposoby wybierania znaczących minimów:

1. "Zwykła" transformacja wododziałowa wykonywana jest na obrazie gradientowym g_k na poziomie segmentacji k. Minima lokalne z obrazu gradientowego g_s poziomu inicjującego s są przenoszone na poziom k, gdzie stają się odpowiednio większymi obszarami. Następnie z obrazu gradientowego g_k wybierane są te minima lokalne, których zlewisko ma przynajmniej (uzyskane podczas "zwykłej" transformacji) jeden punkt wspólny z minimum przeniesionym z poziomu s. Wybrane minima służą jako markery przy kolejnym wyznaczaniu transformacji wododziałowej obrazu g_k.

2. Podobnie jak w pierwszym rozwiązaniu minima lokalne z obrazu g_s przenoszone są na poziom k, jednak za istotne minima obrazu g_k uznawane są, te które znajdują się na większych obszarach wyznaczonych przez minima przeniesionych z poziomu s.

Według autorów [33] drugie rozwiązanie daje lepsze wyniki.

Praca [95] prezentuje inne podejście do problemu przenoszenia etykiet między poziomami piramidy. Zlewiska z obrazu w pełnej rozdzielczości są powiązane z minimami lokalnymi. Każde z tych minimów jest przenoszone na obraz niższej rozdzielczości, a następnie z każdego przeniesionego minimum jest wyznaczana ścieżka najszybszego spadku prowadząca do minimum lokalnego obrazu gradientowego niższej rozdzielczości. W ten sposób łączone są minima ze wszystkich poziomów piramidy. Otrzymana hierarchia umożliwia przeniesienie etykiet z segmentacji obrazu najniższej rozdzielczości na zlewiska otrzymane dla obrazu oryginalnego.

W pracy [124] cały proces zaczyna się od segmentacji obrazu najmniejszej rozdzielczości. Następnie etykiety zlewisk otrzymane w wyniku transformacji wododziałowej przenoszone są na kolejny, większy, obraz w hierarchii. Przenoszenie etykiet ma miejsce tylko w punktach, w których moduł gradientu obrazu wyższej rozdzielczości wynosi 0. Przeniesione etykiety umożliwiają przeprowadzenie transformacji wododziałowej na obrazie większej rozdzielczości, ale tylko w punktach, w których moduł gradientu jest większy od 0. Dzięki takiemu postępowaniu zlewiska związane z różnymi minimami obrazu wyższej rozdzielczości mogą mieć identyczne etykiety. Powyższy schemat jest powtarzany tak długo, aż nie zostanie utworzony obraz pełnej rozdzielczości. Mimo że w tym rozwiązaniu transformacji wododziałowej poddawane są wszystkie obrazy, to otrzymywana hierarchia powiązań między minimami poszczególnych poziomów bardzo przypomina tę z [95].

Drugą strategią tworzenia powiązań między poziomami piramidy jest przeprowadzenie transformacji wododziałowej wszystkich jej obrazów składowych i

szukanie związków między otrzymanymi zlewiskami. Przykładem może być [57] gdzie transformacji wododziałowej poddawany jest każdy poziom piramidy obrazów, jednak tutaj za każdym razem wykonywana jest niezmodyfikowana wersja transformacji. Dopiero po otrzymaniu wszystkich segmentacji, regiony z kolejnych poziomów są łączone. Zlewiska uzyskane dla mniejszego obrazu są odpowiednio skalowane, a następnie ich etykiety są przypisywane tym zlewiskom z obrazu większej rozdzielczości. Decyzja o tym, którą etykietę przypisać danemu zlewisku, jest podejmowana na podstawie liczby wspólnych pikseli.

Również w rozwiązaniu zaproponowanym w [52] na każdym poziomie piramidy wykonywana jest transformacja wododziałowa, za pomocą której uzyskiwane są linie działów wodnych. Dla najniższego poziomu (najmniejszego obrazu) oprócz linii wododziałowych generowany jest również obraz uproszczony zawierający średnie wartości punktów w poszczególnych zlewiskach oraz obraz z etykietami. W celu przeniesienia wyników segmentacji na obraz w wyższej rozdzielczości, na obrazie uproszczonym (ze średnimi zlewisk) wykonywany jest jeden krok odwrotnej transformaty falkowej. Na podstawie otrzymanego obrazu, wyniku transformacji wododziałowej dla obrazu wyższej rozdzielczości i obrazu z etykietami ustalane są etykiety dla zlewisk na bieżącym poziomie. Wyniki przenoszone są między poziomami, aż do uzyskania segmentacji obrazu oryginalnego.

Trzeci typ strategii łączenia obrazów w różnej rozdzielczości zaprezentowano w pracach [48, 49]. Po utworzeniu piramidy obrazów, transformacji wododziałowej poddawany jest tylko obraz najniższej rozdzielczości. Dla wszystkich klas obliczane są średnie zlewisk, które zastępują etykiety w wyniku segmentacji. Na liniach wododziałowych umieszczane są oryginalne wartości pikseli. Następnie w obrazach szczegółowych zawierających informacje o wyższych częstotliwościach zerowane są piksele, które nie odpowiadają liniom wododziałowym. Po przeprowadzeniu jednego kroku odwrotnej transformacji wododziałowej otrzymywany jest obraz, w którym należy przypisać piksele leżące między zlewiskami do najbliższych klas. Wewnętrzne części zlewisk nie są zmieniane dzięki wyzerowaniu odpowiednich współczynników. Poprzez powtarzanie powyższej procedury możliwe jest przeniesienie wyników segmentacji zmniejszonego obrazu do pełnej rozdzielczości.

Kolejnym stosunkowo nowym sposobem usuwania nadsegmentacji jest wykorzystanie, razem z transformacją wododziałową, innej metody segmentacji. Może nią być na przykład metoda *aktywnych konturów* (ang. *snakes, active contours*) [30, 129], w której kontur początkowy ewoluuje pod wpływem różnych sił (zewnętrznej i wewnętrznej), tak aby osiągnąć minimum energii. Siła wewnętrzna zapobiega rozciąganiu i wyginaniu konturu (co umożliwia uzyskanie gładkich krawędzi), natomiast siła zewnętrzna (często związana z obrazem gradientowym) powoduje jego przyciąganie do krawędzi segmentowanego obiektu [129]. Jednym ze sposobów na powiązanie transformacji wododziałowej z metodą aktywnych

konturów jest użycie dwóch metod po kolei. Jako pierwsza wykonywana jest transformacja wododziałowa. Wymagany kontur startowy jest rysowany ręcznie. Następnie jest on przyciągany do najbliższych krawędzi zlewiska leżących wewnątrz obrysowanego obszaru i przeprowadzana jest ewolucja aktywnego konturu niezależnie do linii działów wodnych [129].

Podobny sposób zaprezentowano w [30]. Jednak w tej pracy wykorzystano *model balonu* (ang. *baloon model*) [27], albo inaczej "nadmuchiwane" aktywne kontury. Różnią się one od typowych tym, że oprócz siły zewnętrznej i wewnętrznej działa na nie dodatkowa siła, która powoduje rozszerzanie lub zmniejszanie konturu (w zależności od znaku). Rozwiązuje to problem inicjalizacji procesu ewolucji. Dzięki dodatkowej sile kontur początkowy nie musi się znajdować bardzo blisko rzeczywistej krawędzi segmentowanego obiektu. W [30] wykorzystano tę właściwość. Po wykonaniu transformacji wododziałowej wybierane jest zlewisko, które należy do segmentowanego obiektu. Jego krawędź staje się początkowym konturem, który ewoluuje tak długo, aż zostanie dopasowany do poszukiwanej krawędzi obiektu. Jak widać użycie "nadmuchiwanych" aktywnych konturów pozwoliło wyeliminować potrzebę ręcznego rysowania konturu początkowego.

Inną metodą segmentacji jaką można zastosować do grupowania zlewisk jest progowanie. Autor niniejszej pracy zaproponował w [103] modyfikację szybkiego algorytmu progowania [125] przystosowującą go do pracy na zlewiskach. Podczas tworzenia podziału na wiele klas korzysta on z algorytmu optymalnego progowania dzielącego określony fragment obrazu na dwie części. W [125] optymalne progowanie dzieli wartości na te poniżej i powyżej progu, w [103] zlewiska należące do wybranego fragmentu są dzielone na te, które mają średnią poniżej i te które mają średnią powyżej optymalnego progu. Oznacza to, że w drugim przypadku pewne odcienie szarości mogą występować w różnych klasach – zlewiska są traktowane jako całość i nie mogą być podzielone. Pierwszym etapem grupowania zlewisk w zadaną liczbę klas jest optymalne progowanie całego zakresu wartości średnich. Znajdowana jest wartość progowa, która dzieli zlewiska na dwie grupy. Gdy zostanie ustalony pierwotny podział, rozpoczyna się iteracyjna część procesu. Na podstawie histogramów poszczególnych klas podejmowana jest decyzja, którą klasę należy podzielić. Jest ona dzielona na dwie części za pomocą algorytmu optymalnego progowania. W dalszej kolejności przeprowadzana jest optymalizacja wszystkich istniejących progów – każdy próg t_n zastępowany jest nowym progiem t'_n – znalezionym w wyniku optymalnego progowania przeprowadzonego w zakresie (t_{n-1}, t_{n+1}). Po modyfikacji wszystkich progów obliczana jest wartość funkcji oceniającej. Jeżeli wartość funkcji poprawiła się, to nowe progi są zapamiętywane i ponawiana jest próba optymalizacji progów. Jeśli jednak wartość funkcji oceniającej jest gorsza, algorytm przechodzi do następnej iteracji i dodaje nową klasę. Powyższy schemat powtarzany jest do osiągnięcia zadanej liczby klas.

Kolejną kategorią sposobów redukcji nadsegmentacji jest modyfikacja samego algorytmu transformacji wododziałowej. Mimo różnic modyfikacje zazwyczaj dotyczą łączenia zlewisk podczas symulacji zalewania rzeźby terenu. W standardowym algorytmie zlewiska nigdy się ze sobą nie łączą. Dopuszczenie, pod pewnymi warunkami, do ich łączenia prowadzi do zmniejszenia nadsegmentacji.

W pracy [7] liczba zlewisk zmniejszana dzięki podzieleniu procesu zalewania na dwa etapy. W pierwszym rzeźba terenu zalewana jest do określonego poziomu, przy czym łączenie się zlewisk jest dopuszczone. Łączone są wszystkie zlewiska, które się stykają. Efektywnie oznacza to zmodyfikowanie segmentowanej powierzchni. Drugi etap jest taki sam jak w "tradycyjnej" transformacji wododziałowej i prowadzi do wyznaczenia linii działów wodnych.

Autorzy pracy [118] zaproponowali nową wersję algorytmu realizującego transformację wododziałową przez symulowane zanurzanie. Próby przeprowadzono na trójwymiarowych obrazach medycznych. Zmniejszenie nadsegmentacji osiągnięto poprzez dodanie możliwości wyboru kroku, z jakim postępuje zalewanie terenu. W standardowym algorytmie w kolejnych krokach poziomy różniące się wysokością o 1 zalewane są pojedynczo, a nowe minima wykryte na każdym z nich otrzymują nowe etykiety. W algorytmie zaprezentowanym w [118] minima zalewane w trakcie przetwarzania bieżącego przedziału (o wysokości równej krokowi) są łączone z bieżącymi zlewiskami, jeżeli się z nimi stykają i nie otrzymują nowych etykiet. Nowe etykiety są nadawane tylko minimom, które nie stykają się z istniejącymi zlewiskami. Stopień redukcji nadsegmentacji można regulować poprzez zmianę kroku zalewania.

W [4] rozszerzono transformację wododziałową, tak aby łączenie zlewisk podczas symulacji zalewania nie zawsze było blokowane. W trakcie pracy algorytmu monitorowana jest głębokość, powierzchnia i objętość zlewisk (lub tylko niektóre z tych parametrów) i na ich podstawie zapada decyzja o tym, czy dopuścić do połączenia tego zlewiska z innym, czy też zablokować połączenie.

Ostatnią kategorią metod zmniejszania nadsegmentacji są rozwiązania opracowane pod kątem segmentacji obrazów określonego typu. Wtedy możliwe jest wykorzystanie cech charakterystycznych segmentowanych obiektów do usuwania nadsegmentacji. W [2] przedstawiono wykorzystanie transformacji wododziałowej do segmentacji zbioru obrazów przedstawiających różne warstwy tkanki. Obrazy uzyskano za pomocą mikroskopu konfokalnego. W celu zmniejszenia nadsegmentacji, korzystając z tego, że histogram przetwarzanych obrazów jest unimodalny, obraz jest progowany, w celu usunięcia najmniejszych wartości. Po wykonaniu transformacji wododziałowej do łączenia zlewisk wykorzystywany jest fakt, że segmentowane komórki mają określony rozmiar, a więc jeżeli region ma rozmiar mniejszy niż określony próg i równocześnie graniczy z jednym (lub więcej) dużym regionem to jest łączony z tym, który ma największą powierzchnię. Po wykonaniu połączenia nowe zlewisko jest uwzględniane jako kandydat do kolejnego łączenia,

dopiero po sprawdzeniu innych możliwości. Małe zlewiska bezpośrednio stykające się z dużymi regionami są uwzględniane podczas łączenia w pierwszej kolejności. A te, które nie mają dużych sąsiadów są łączone ze sobą. Jeżeli po połączeniu ich rozmiar przekracza zadany próg, to połączone regiony są uznawane za komórkę w przeciwnym wypadku są odrzucane.

Inne rozwiązanie wymagające znajomości charakterystyki segmentowanego obrazu zaprezentowano w [37]. W trakcie obliczania transformacji wododziałowej (algorytmem symulującego zanurzanie) tworzona jest hierarchia (drzewo), opisująca połączenia zlewisk, które miałyby miejsce, gdyby przelewanie się wody między zlewiskami było dopuszczone. Zlewiskom zawsze odpowiadają liście drzewa. W momencie łączenia nadrzędnym węzłem staje się ten, który odpowiada regionowi o większej powierzchni. Nowe węzły w drzewie tworzone są, tylko gdy węzłem nadrzędnym miałby stać się liść. Drzewo otrzymane za pomocą, opisanego powyżej, zmodyfikowanego algorytmu transformacji wododziałowej jest dopasowywane do modelu na podstawie funkcji kosztu, tak aby koszt został zminimalizowany. Funkcja ta jest obliczana dla charakterystyk dostępnych w modelu i każdemu z dostępnych parametrów przyporządkowuje koszt z zakresu $[0; 1]$ w zależności od tego, jak bardzo różni się od modelu. Koszty dla wszystkich klas i parametrów są sumowane. Po dopasowaniu drzewa segmentacji do modelu, poszczególnym zlewiskom nadawane są etykiety, takie jak zapisane w modelu. Jak widać zastosowanie metody zaprezentowanej w [37] wymaga wcześniejszego utworzenia wzorca segmentacji na przykład dla obrazu pochodzącego z rezonansu magnetycznego i przedstawiającego skan głowy. Model tworzony jest na podstawie ręcznej segmentacji opracowanej przez eksperta i drzewa uzyskanego dla wzorcowego obrazu w taki sam sposób, jak podczas automatycznej segmentacji. Etykiety nadane przez eksperta są przypisywane do poszczególnych zlewisk. Struktura drzewa zostaje uproszczona (węzły z identycznymi etykietami są łączone. Następnie dla węzłów modelu obliczane są określone charakterystyki i ustalane tolerancje.

Rozdział 3

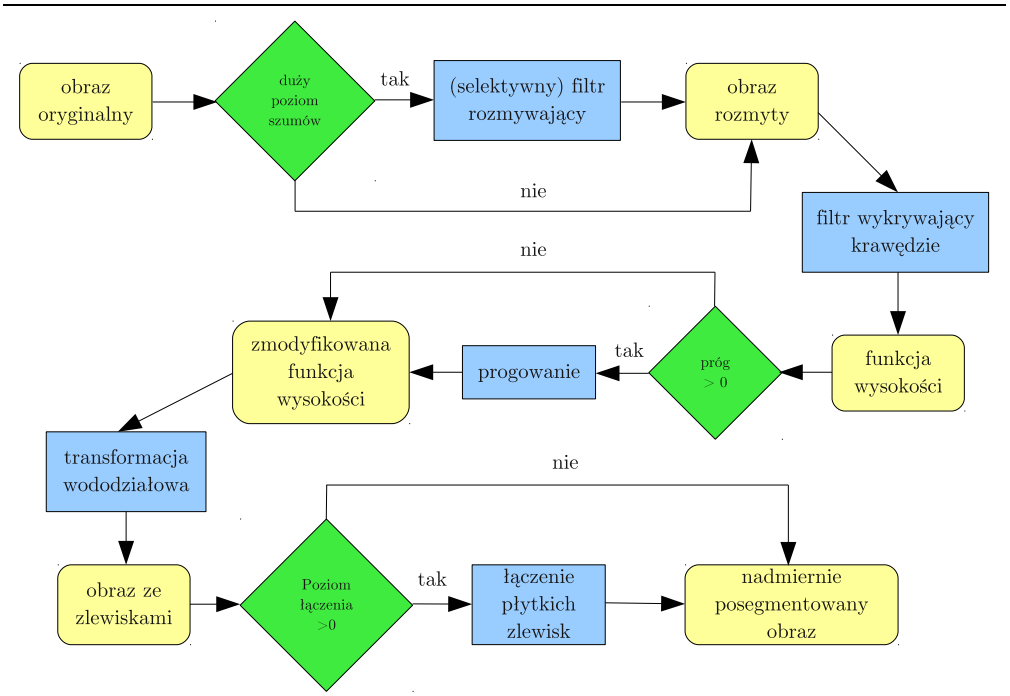
Redukcja nadsegmentacji

3.1 Wprowadzenie

Jak wspomniano wcześniej wyniki otrzymane za pomocą transformacji wododziałowej są w znaczącym stopniu nadmiernie posegmentowane. Rysunek 2.7 wyraźnie pokazuje, że użyta samodzielnie transformacja wododziałowa często nie daje satysfakcjonujących rezultatów. Metody redukcji nadsegmentacji omówione w punkcie 2.7 są często projektowane z myślą o rozwiązaniu konkretnego zadania. Stwarza to problemy przy próbie ich innego zastosowania. Powstają także uniwersalne metody segmentacji obrazów, które nie zawsze pozwalają na uzyskanie zamierzonych wyników. Dlatego autor niniejszej pracy zaproponował [104] rozwiązanie, które można łatwo zaadaptować do osiągnięcia konkretnego celu segmentacji. W początkowej fazie tej metody obraz jest przygotowywany, wyznaczana jest transformacja wododziałowa i przeprowadzana jest redukcja nadsegmentacji, która odbywa się w dwóch etapach. Wstępna redukcja polega na odpowiednim filtrowaniu obrazu oryginalnego i gradientowego, a także na łączeniu płytkich zlewisk z ich sąsiadami. Ostateczna redukcja nadsegmentacji odbywa się przy użyciu hierarchicznych metod analizy skupień, które ze względu na swoją modułową konstrukcję, mają możliwość łatwej adaptacji do różnych typów obrazów i zadań.

3.2 Przygotowanie obrazu i wstępna redukcja nadsegmentacji

Przedstawione rozwiązanie składa się z kilku etapów. Pierwszym z nich jest właściwe przygotowanie obrazu. Zastosowano następujący schemat przetwarzania obrazu przed zastosowaniem analizy skupień:



Rysunek 3.1: Schemat przedstawiający sposób uzyskania nadmiernie posegmentowanego obrazu, który stanowi dane wejściowe dla analizy skupień.

1. Zastosowanie selektywnego filtru rozmywającego do segmentowanego obrazu.
2. Wyznaczenie obrazu gradientowego.
3. Progowanie obrazu gradientowego, w celu usunięcia najmniejszych wartości gradientu.
4. Przeprowadzenie transformacji wododziałowej.
5. Wstępna redukcja nadsegmentacji poprzez łączenie płytkich zlewisk.

Powyższy schemat został przedstawiony na rysunku 3.1. Poniżej zostanie wykazane również, że rzeczywiste obrazy przetworzone w ten sposób w dalszym ciągu są nadmiernie posegmentowane dlatego konieczne jest dalsze przetwarzanie wyników w celu uzyskania satysfakcjonujących rezultatów.

W punkcie 3.2.1, omówiono pokrótce wykorzystywane filtry rozmywające. Punkt 3.2.2 omawia i prezentuje skutki progowania obrazu gradientowego. Punkt 3.2.3, natomiast, dotyczy algorytmu wstępnego łączenia płytkich zlewisk. Opisane metody wstępnej redukcji nadsegmentacji nadają się do użycia zarówno z obrazami monochromatycznymi jak i barwnymi.

3.2.1 Zastosowanie selektywnego filtru rozmywającego

W pracy zastosowano dwa selektywne filtry rozmywające, których zasada działania opiera się na rozwiązywaniu cząstkowego równania różniczkowego. Filtry tego typu zdobyły dużą popularność ponieważ skutecznie eliminują szum lecz dobrze zachowują istotne elementy obrazu [35, 37, 45, 64, 80, 93, 120, 123, 129]. W procesie segmentacji przedstawionym na rysunku 3.1 mogą być stosowane zamiennie w bloku oznaczonym jako *selektywny filtr rozmywający.*

Pierwszy z filtrów został zaproponowany przez P. Peronę i J. Malika i wykorzystuje równanie dyfuzji anizotropowej postaci [80]:

$$\frac{\partial I}{\partial t} = div[c(x,y,t)\nabla I] = c(x,y,t)\Delta I + \nabla c \cdot \nabla I \tag{3.1}$$

Różni się ono od równania dyfuzji izotropowej $\frac{\partial I}{\partial t} = c\Delta I$ tym, że czynnik c nie jest stałą lecz funkcją. Dzięki temu różne obszary obrazu rozmywane są w różnym stopniu. Przyjęcie, że (1) wartość c zależy od modułu gradientu tzn.:

$$c(x,y,t) = f\left(\|\nabla I(x,y,t)\|\right) \tag{3.2}$$

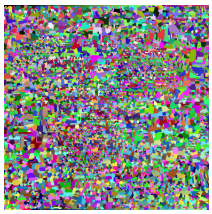
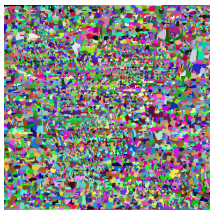
i (2), że f jest nieujemną, monotonicznie malejącą funkcją, taką że $f(0) = 1$, gwarantuje, że dyfuzja będzie zachodzić głównie we wnętrzu jednorodnych regionów i nie będzie dotyczyć krawędzi (przy odpowiednio dobranych parametrach może nawet wzmacniać krawędzie) [80]. Perona i Malik zaproponowali dwie różne funkcje f postaci:

$$f(\nabla I) = e^{-\left(\frac{\|\nabla I\|}{K}\right)^2} \text{ lub } f(\nabla I) = \frac{1}{1 + \left(\|\nabla I\|/K\right)^2}, \tag{3.3}$$

gdzie K to stała nazywana konduktancją. Jej wpływ na proces dyfuzji zależy od rodzaju zastosowanej funkcji f jednak zazwyczaj większe wartości K zwiększają jego intensywność, czyli powodują większe rozmycie obrazu. Pierwsza z funkcji f lepiej zachowuje krawędzie o wysokim kontraście (tę wykorzystano w niniejszej pracy), a druga szerokie regiony. Istnieje także wiele innych propozycji funkcji f [120]. W [35, 93] przedstawiono sposoby dostosowania tego filtru do obrazów, w których piksele opisane są wektorami. W [35] zaproponowano połączenie danych o dwóch obrazach gradientowych obliczonych dla dwóch obrazów MRI. Można je traktować jak jeden posiadający wiele składowych, ponieważ otrzymano je równocześnie, w trakcie jednego badania. Informacja o module gradientu z dwóch obrazów zamieniana jest na współczynnik dyfuzji w następujący sposób:

$$c(x,y,t) = f\left(\sqrt{\|\nabla I_1(x,y,t)\|^2 + \|\nabla I_2(x,y,t)\|^2}\right). \tag{3.4}$$

W [93] opisano rozszerzenie tego filtru na obrazy barwne. W tej pracy jednak informacje z różnych składowych połączono dzięki nowemu sposobowi obliczania gradientu dla obrazów barwnych (opisanego w punkcie 2.5.2).

kwiatek_3	**sufit_2**	**kwiatek_3**	**sufit_2**
			
(a) z filtrem 3.1: *9692* zlewiska	(b) z filtrem 3.1: *13941* zlewisk	(c) z filtrem 3.5: *10084* zlewiska	(d) z filtrem 3.5: *14668* zlewisk

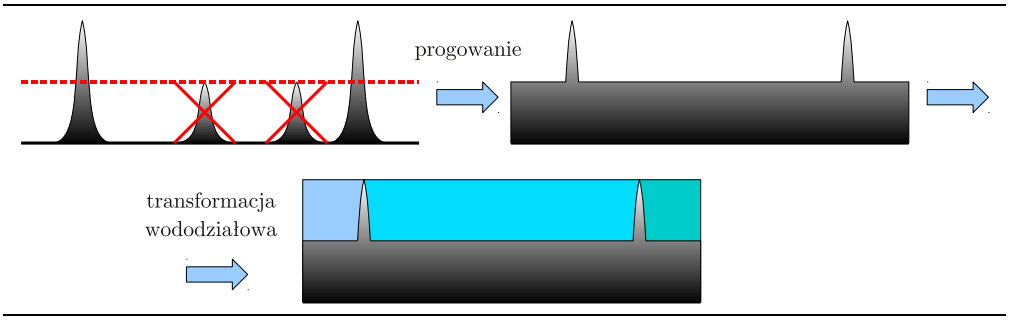
Rysunek 3.2: Wpływ zastosowania selektywnego filtru rozmywającego (opartego na równaniu 3.1 lub 3.5) do obrazu oryginalnego na nadsegmentację. We wszystkich przypadkach użyto następujących parametrów: liczba iteracji $n = 5$, $\Delta t = 0,12$, $K = 0,3$. Oryginalne obrazy *kwiatek_3* i *sufit_2* znajdują się na rysunku 2.7.

Drugi filtr został zaproponowany przez R. Whitakera i X. Xue [123]. Jest on oparty na popularnej metodzie poziomic (ang. *level-set*) [26, 45, 64, 123]. W metodzie tej równanie różniczkowe opisuje ruch krzywej (poziomicy odpowiadającej na przykład stałej jasności), która porusza się z prędkością zależną od swojej krzywizny. Zastosowanie metody poziomic do skonstruowania selektywnego filtru rozmywającego wymaga modyfikacji, ponieważ krzywe poruszające się ruchem opisanym oryginalnym równaniem, z początku dążą do bardziej gładkiego kształtu (o mniejszej krzywiźnie), a następnie zapadają się i znikają [64]. Filtr rozmywający zastosowany w pracy wykorzystuje równanie różniczkowe postaci [123]:

$$\frac{\partial I}{\partial t} = |\nabla I|\kappa_c; \quad \kappa_c = div\left[c(|\nabla I|)\frac{\nabla I}{|\nabla I|}\right], \tag{3.5}$$

gdzie κ_c to zmodyfikowany współczynnik krzywizny. Modyfikacja polega na wprowadzeniu czynnika zależnego od modułu gradientu $c(|\nabla I|)$ i powoduje zmniejszenie prędkości poziomic w miejscach, w których moduł gradientu obrazu ma dużą wartość. W wyniku tego małe obszary (czyli te o dużej krzywiźnie krawędzi), nie różniące się znacząco od swoich sąsiadów (niewielki gradient) są rozmywane szybko. Najwolniej natomiast rozmywane są krawędzie dużych obszarów o dużym kontraście. Również ten filtr można rozszerzyć na obrazy barwne [45, 123].

Rysunek 3.2 przedstawia jaki wpływ na nadsegmentację ma zastosowanie filtru rozmywającego do obrazów testowych z rysunku 2.7. Parametry filtrów były takie same. W przypadku obrazu *kwiatek_3* redukcja nadsegmentacji jest mniejsza (około trzy tysiące zlewisk) niż w wypadku obrazu *sufit_2* (około sześć tysięcy). Należy zaznaczyć, że w przypadku drugiego z obrazów nadsegmentacja była dużo większa, ze względu na wyraźny szum występujący w obrazie. Oba filtry osiągnęły



Rysunek 3.3: Ilustracja wstępnego zmniejszania nadsegmentacji poprzez progowanie obrazu gradientowego. Wartości mniejsze od progu są zastępowane jego wartością.

podobne wyniki w redukcji nadsegmentacji jednak nie pozwoliły na sprowadzenie liczby regionów do satysfakcjonujących poziomów.

3.2.2 Progowanie obrazu gradientowego

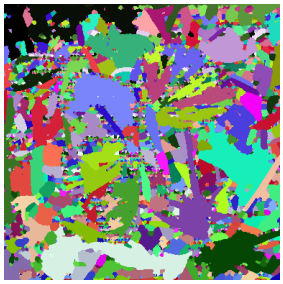
Kolejną metodą redukcji nadsegmentacji jest progowanie obrazu gradientowego. W niniejszej pracy stanowi ono drugi, po rozmywaniu, etap wstępnej redukcji nadsegmentacji (rysunek 3.1). Zastosowano prostą metodę progowania, która w obrazie gradientowym zastępuje wszystkie wartości mniejsze niż próg t wartością tego progu. Jest on wyznaczany za pomocą wzoru:

$$t = min + t_p \cdot (max - min), \tag{3.6}$$

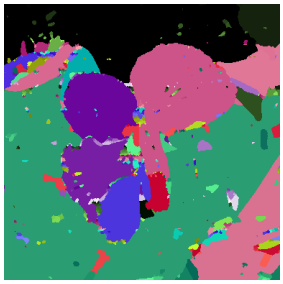
gdzie t_p to próg wyrażony jako procentowa wartość różnicy między minimalną a maksymalną wartością występującą w obrazie gradientowym, a min i max to odpowiednio wartość minimalna i maksymalna występująca w obrazie gradientowym. Rysunek 3.3 ilustruje wpływ progowania na nadsegmentację. W dalszej części pracy wartości progu podawane będą jako wartości procentowe.

Na rysunku 3.4 przedstawiono wpływ progowania na stopień nadsegmentacji. Rysunki 3.4a oraz 3.4d pokazują wynik zastosowania prawidłowego, niewielkiego progu. Nadsegmentacja została zredukowana z około 10000 (dla obrazu *kwiatek_3*) i 14500 (dla obrazu *sufit_2*) odpowiednio do poziomu około 3500 i 3800 zlewisk. Obrazy są nadal nadmiernie posegmentowane, jednak nie zostały w nich połączone regiony związane z różnymi obiektami. Obrazy 3.4b, 3.4c, 3.4e oraz 3.4f prezentują skutki zastosowania zbyt wysokiego progu. Liczba zlewisk w obrazach testowych jest nadal wyraźnie większa niż liczba obiektów znajdujących się na obrazie, podczas gdy różne klasy zostały połączone. W obrazie *sufit_2* połączone zostały różne elementy obrazu w prawym górnym rogu, a w obrazie *kwiatek_3* większość elementów tła stała się jednym regionem. Pokazuje to, że także zastosowanie filtru rozmywającego w połączeniu z progowaniem obrazu gradientowego nie umożliwia osiągnięcia satysfakcjonującej redukcji nadsegmentacji.

kwiatek_3



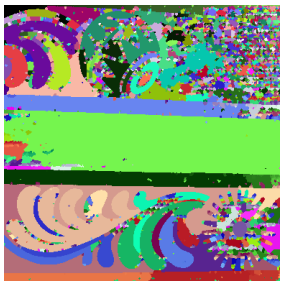
(a)
filtr rozmywający i próg 5%: *3489* zlewisk



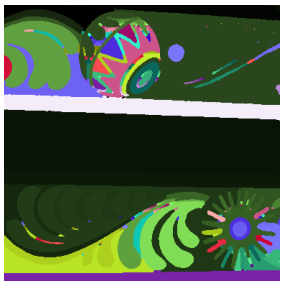
(b)

(c)
filtr rozmywający i próg 10%: *970* zlewisk (zlewiska oraz średnie zlewisk)

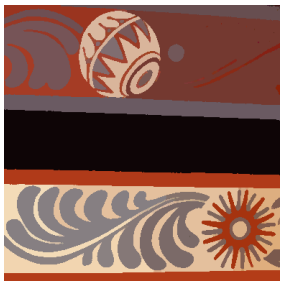
sufit_3



(d)
filtr rozmywający i próg 5%: *3774* zlewiska



(e)



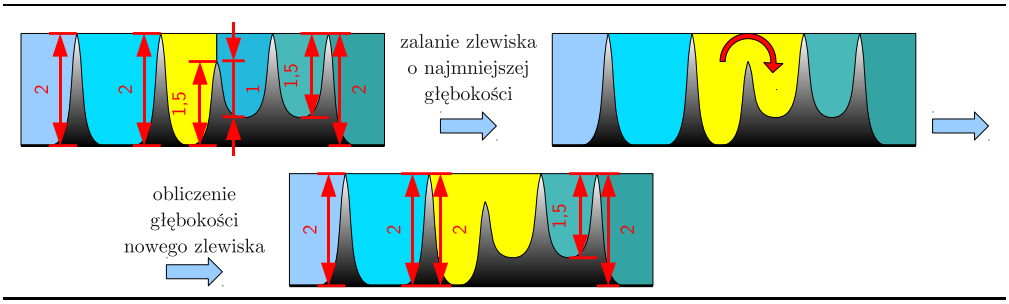
(f)
filtr rozmywający i próg 15%: *468* zlewisk (zlewiska oraz średnie zlewisk)

Rysunek 3.4: Wpływ progowania obrazu gradientowego na nadsegmentację. Obrazy (b) i (c) oraz (e) i (f) przedstawiają skutki zastosowania zbyt wysokiego progu.

3.2.3 Zalewanie płytkich zlewisk

Ostatnim – trzecim – etapem wstępnej redukcji nadsegmentacji, stosowanym w niejszej pracy, jest łączenie sąsiednich zlewisk na podstawie ich głębokości [45]. Łączenie odbywa się, zgodnie z diagramem przedstawionym na rysunku 3.1 po wyznaczeniu transformacji wododziałowej. Zalewane przez sąsiadów o większej głębokości są tylko płytkie zlewiska, czyli takie, których głębokość nie jest większa niż *poziom łączenia* definiowany jako procent różnicy między maksymalną i minimalną wartością występującą w obrazie. Rysunek 3.5 przedstawia sposób łączenia płytkich zlewisk. Ilustruje on trzy kroki konieczne do połączenia dwóch zlewisk: (1) wybranie kandydatów do połączenia, (2) wykonanie połączenia zlewisk – zmiana etykiet, (3) obliczenie głębokości nowego zlewiska i uaktualnienie listy potencjalnych połączeń.

Poniżej przedstawiono szczegółowy opis algorytmu łączenia płytkich zlewisk. Został on podzielony na trzy etapy: (1) tworzenie tabeli zlewisk, (2) tworzenie po-



Rysunek 3.5: Ilustracja zmniejszania nadsegmentacji poprzez zalewanie płytkich zlewisk. Zlewisko jest zalewane przez sąsiada, z którym dzieli najniższą krawędź, a jego głębokość nie przekracza poziomu łączenia. Na rysunku zlewisko oznaczone kolorem żółtym zalewa swojego płytkiego sąsiada.

czątkowej listy możliwych połączeń, (3) zalewanie płytkich regionów. W tabeli 3.1 został zamieszczony spis użytych oznaczeń wraz z objaśnieniami.

Tabela 3.1: Zestawienie dodatkowych oznaczeń użytych w algorytmie wstępnej redukcji nadsegmentacji

Metoda/pole	**Opis**
obiekt: jkTabRown – przechowuje pary równoważnych etykiet zlewisk	
`aktualnaEtykieta(` *`etykieta`*`)`	Określa aktualną etykietę zlewiska. Jeżeli nie ma etykiet równoważnych, metoda zwraca swój argument.
`dodaj(`*`etykieta1`*`,` *`etykieta2`*`)`	Dodaje parę równoważnych etykiet do tabeli. Para etykiet jest uporządkowana.
klasa: Krawedz – reprezentuje informację o sąsiednim zlewisku	
`etykieta`	Etykieta sąsiedniego zlewiska.
`wysokosc`	Wysokość krawędzi.
klasa: Krawedzie – reprezentuje listę sąsiadów zlewiska	
`dodaj(`*`etykieta`*`,` *`wysokosc`*`)`	Dodaje do listy krawędź wspólną ze zlewiskiem o zadanej etykiecie. Ustawia też wysokość krawędzi.
`znajdz(`*`etykieta`*`)`	Szuka krawędzi (obiektu klasy Krawedz) wspólnej ze zlewiskiem o zadanej etykiecie.
klasa: Polaczenie – reprezentuje planowane połączenie dwóch zlewisk	
`glebokosc`	Głębokość zlewiska jest określona jako różnica wysokości najniższej krawędzi i minimum zlewiska.
`zalewajacy`	Etykieta zlewiska zalewającego. Po połączeniu nowe zlewisko ma tę etykietę.
`zalewany`	Etykieta zlewiska zalewanego.

ciąg dalszy na następnej stronie...

Tabela 3.1 – ciąg dalszy

Metoda/pole	**Opis**
obiekt: roboczePolaczenia / polaczenia – przechowuje uporządkowaną (wg. głębokości zlewisk) listę planowanych połączeń płytkich zlewisk	
`dodaj(`*`zalewany, zalewajacy, poziomLaczenia`*`)`	Dodaje do listy połączeń informację o potencjalnym połączeniu zlewisk. Zapisywane są etykiety dwóch zlewisk oraz poziom, przy którym zostanie ono zalane.
`poczatek()`	Zwraca obiekt klasy Polaczenie opisujący zalewanie zlewiska o najmniejszej głębokości.
`pusta()`	Zwraca informację, czy lista połączeń jest pusta.
`usunPoczatek()`	Usuwa obiekt klasy Polaczenie, opisujący zalewanie zlewiska o najmniejszej głębokości.
klasa: Zlewisko – reprezentuje zlewisko	
`krawedzie`	Lista krawędzi (i sąsiadów) zlewiska.
`minimum`	Minimalna wartość piksela w zlewisku.
`xMin`	Położenie piksela o minimalnej wartości.
obiekt: zlewiska – przechowuje bieżącą listę zlewisk	
`dodaj(`*`etykieta, minimum, polozenie`*`)`	Dodaje do spisu nowe zlewisko o zadanej etykiecie, minimalnej wartości i położeniu.
`operator[`*`etykieta`*`]`	Zwraca obiekt, opisujący zlewisko o zadanej etykiecie.
`polacz(`*`zrodlowe, docelowe, tabela`*`)`	Łączy (tylko w spisie) dwa zlewiska. Nowe zlewisko ma etykietę taką samą jak zlewisko docelowe, mniejsze z dwóch minimów oraz połączone listy krawędzi. Krawędzie wewnętrzne są usuwane. Listy krawędzi łączone są w taki sposób, że lista wynikowa jest posortowana rosnąco. Zlewisko źródłowe jest kasowane ze spisu. Na podstawie tabeli równoważnych etykiet uaktualnia etykiety sąsiadów zapisanych w krawędziach.
`sortujRosnaco-ListyKrawedzi()`	Sortuje rosnąco listy krawędzi we wszystkich zlewiskach znajdujących się w spisie zlewisk. Krawędzie sortowane są według wysokości.

W pierwszym etapie algorytm 3.1 przegląda wszystkie punkty obrazu (współrzędne są w zbiorze D) otrzymanego w wyniku transformacji wododziałowej (*obrazWy*) i sprawdza, czy etykieta bieżącego punktu istnieje w spisie zlewisk. Jeżeli nie, to dodaje etykietę bieżącego punktu (efektywnie dodaje zlewisko) razem z informacją o wartości punktu (w obrazie gradientowym – *obrazWe*) i jego położeniu (współrzędna x) do spisu *zlewiska*. Jeżeli informacja o zlewisku znajduje

się w spisie i bieżący punkt ma wartość mniejszą niż dotychczasowe minimum, to informacja o minimum (wartość i położenie) jest aktualizowana. Kolejnym krokiem jest sprawdzenie, czy bieżący punkt x leży na granicy zlewisk. Jeżeli tak (co oznacza, że przynajmniej jeden z sąsiednich punktów ma inną etykietę), to krawędź jest dodawana do spisu *krawedzie* (jeżeli nie istniała wcześniej) lub aktualizowana, jeżeli znajdowała się już wcześniej w spisie. Krawędź zawiera informacje o swojej wysokości i etykiecie sąsiada. Wysokość krawędzi to wysokość jej najniższego punktu. Za wysokość konkretnego punktu krawędzi przyjmuje się wartość (w obrazie gradientowym – *obrazWe*) większego z dwóch sąsiadujących ze sobą punktów, należących do różnych zlewisk. Pod koniec etapu listy krawędzi w poszczególnych zlewiskach są sortowane rosnąco według wysokości krawędzi.

Po opisanym powyżej kroku algorytmu istnieje spis wszystkich zlewisk (*zlewiska*), w każdym zlewisku znaleziono minimum (położenie i wartość). Ponadto dla każdego zlewiska utworzona została posortowana lista krawędzi (*krawedzie*) zawierających informację o etykietach sąsiadów i wysokościach tych krawędzi.

Algorytm 3.1 Tworzenie tabeli zlewisk

```
1: for all x ∈ D do
2:   if zlewiska[obrazWy[x]] = NULL then
3:     zlewiska.dodaj(obrazWy[x],obrazWe[x],x)
4:   else if obrazWe[x] < zlewiska[obrazWy[x]].minimum then
5:     zlewiska[obrazWy[x]].minimum ← obrazWe[x]
6:     zlewiska[obrazWy[x]].xMin ← x
7:   end if
8:   for all s ∈ N(x) do
9:     if obrazWy[x] ≠ obrazWy[s] then
10:      max ← obrazWe[x]
11:      if max < obrazWe[s] then
12:        max ← obrazWe[s]
13:      end if
14:      if not zlewiska[obrazWy[x]].krawedzie.znajdz(obrazWy[s])
         ≠ NULL then
15:        zlewiska[obrazWy[x]].krawedzie.dodaj(obrazWy[s], max)
16:      else if zlewiska[obrazWy[x]].krawedzie.znajdz(
         obrazWy[s]).wysokosc > max then
17:        zlewiska[obrazWy[x]].krawedzie.znajdz(obrazWy[s]).wysokosc ←
           max
18:      end if
19:    end if
20:  end for
21: end for
22: zlewiska.sortujRosnacoListyKrawedzi()
```

Drugim etapem łączenia płytkich zlewisk (algorytm 3.2) jest utworzenie początkowej listy połączeń (*roboczePolaczenia*). Przeglądany jest cały spis zlewisk (należy zaznaczyć, że numeracja zlewisk nie musi być ciągła). Po znalezieniu zlewiska obliczana jest jego głębokość i jeżeli nie przekracza ona poziomu łączenia, to do listy połączeń dodawany jest wpis o zalaniu bieżącego zlewiska przez sąsiada. Zalewającym sąsiadem jest, ten który graniczy z bieżącym zlewiskiem przez krawędź o najniższej wysokości (jest on pierwszy na liście sąsiadów/krawędzi zlewiska). Należy zaznaczyć, że lista połączeń jest kolejką priorytetową, w której o porządku decyduje głębokość zalewanego zlewiska.

Po wykonaniu powyższego kroku algorytmu istnieje lista zlewisk, które będą zalane przez swoich sąsiadów. Należy zaznaczyć, że lista jest tylko wartością początkową dla następnego etapu. Połączenie dwóch zlewisk daje nowe zlewisko innej głębokości i liście sąsiadów, co wymaga aktualizacji listy połączeń.

Algorytm 3.2 Tworzenie początkowej listy możliwych połączeń zlewisk

```
1: for z = 0 to maksymalnaEtykieta do
2:   if zlewiska[z] ≠ NULL then
3:     glebokoscRob ← zlewiska[z].krawedzie.poczatek().wysokosc −
       zlewiska[z].minimum
4:     if glebokoscRob ⩽ poziomLaczenia then
5:       roboczePolaczenia.dodaj(z,
         zlewiska[z].krawedzie.poczatek().etSasiada,glebokoscRob)
6:     end if
7:   end if
8: end for
```

Ostatnim etapem jest faktyczne wykonanie połączeń (algorytm 3.3). W kolejnych iteracjach z roboczej listy (*roboczePolaczenia*) odczytywane są kolejne połączenia. Zawsze zalewane jest zlewisko o najmniejszej głębokości na liście. Lista połączeń jest stale uporządkowana, więc zlewisko o najmniejszej głębokości znajduje się zawsze na początku. Etykiety zalewanego i zalewającego zlewiska są aktualizowane dzięki tabeli równoważności (mogły się zmienić w wyniku połączeń wykonanych w poprzednich iteracjach). Algorytm sprawdza, czy zlewiska nie zostały wcześniej połączone, oraz czy zalewane zlewisko nie jest zalewane po raz drugi. Jeżeli połączenie jest prawidłowe, to jest ono dodawane do ostatecznej listy połączeń (*polaczenia*). Elementy spisu zlewisk są łączone (nowe zlewisko otrzymuje etykietę zlewiska zalewającego, łączone są też listy krawędzi oraz wybierane jest mniejsze z dwóch minimów). Etykiety połączonych zlewisk dodawane są do tabeli równoważnych etykiet. Nowo powstałe zlewisko również może zostać zalane, dlatego następnym krokiem jest utworzenie i dodanie odpowiedniego wpisu do roboczej listy połączeń. Wpis dodawany jest tylko wtedy, gdy głębokość nowego zlewiska nie jest większa niż poziom łączenia. Robocza lista połączeń jest przetwarzana do momentu, w którym nie ma już na niej dostatecznie płytkich

zlewisk.

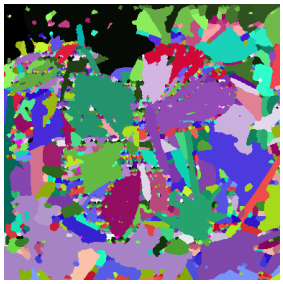
Efektem tego etapu jest powstanie ostatecznej listy połączeń i tabeli równoważnych etykiet. Obie struktury umożliwiają ujednolicenie etykiet połączonych zlewisk w obrazie wynikowym.

Algorytm 3.3 Zalewanie płytkich regionów

```
1: polaczenie ← roboczePolaczenia.poczatek()
2: while not roboczePolaczenia.pusta() and
   polaczenie.glebokosc ⩽ poziomLaczenia do
3:   roboczePolaczenia.usunPoczatek()
4:   zalewany ← jkTabRown.aktualnaEtykieta(polaczenie.zalewany)
5:   zalewajacy ← jkTabRown.aktualnaEtykieta(polaczenie.zalewajacy)
6:   if zalewany ≠ zalewajacy and zalewany ≠ polaczenia.zalewany then
7:     polaczenia.dodaj(zalewany,zalewajacy,polaczenie.glebokosc)
8:     zlewiska.polacz(zalewany,zalewajacy,jkTabRown)
9:     jkTabRown.dodaj(zalewany,zalewajacy)
10:    zalewany ← zalewajacy
11:    zalewajacy ← jkTabRown.aktualnaEtykieta(
       zlewiska[zalewany].krawedzie.poczatek().etSasiada)
12:    glebokoscRob ←
       zlewiska[zalewany].krawedzie.poczatek().wysokosc −
       zlewiska[zalewany].minimum
13:    if glebokoscRob ⩽ poziomLaczenia then
14:      roboczePolaczenia.dodaj(zalewany,zalewajacy,glebokoscRob)
15:    end if
16:  end if
17:  polaczenie ← roboczePolaczenia.poczatek()
18: end while
19: obrazWy.ujednolicenieEtykiet(jkTabRown)
```

Rysunek 3.6 przedstawia wyniki uzyskane w praktyce za pomocą algorytmu zalewania płytkich zlewisk. Jak widać łączenie płytkich zlewisk pozwoliło na dalszą redukcję nadsegmentacji z około 3500 – 3800 do około 2000 zlewisk. Płytkie zlewiska łączono po uprzedniej redukcji nadsegmentacji za pomocą filtru rozmywającego i progowania. Podobnie jak w przypadku progowania najlepsze wyniki przynosi stosowanie niewielkich poziomów łączenia. Rysunki 3.6a i 3.6d przedstawiają wyniki uzyskane przy poziomie 5%. Nadsegmentacja jest w dalszym ciągu wyraźna jednak nie zostały połączone regiony należące do różnych klas. Rysunki 3.6b, 3.6c, 3.6e oraz 3.6f przedstawiają wynik zastosowania zbyt wysokiego poziomu łączenia. Wyniki są podobne do wyników uzyskanych z zastosowaniem zbyt wysokiego progu. Liczba regionów w uzyskanych segmentacjach wynosi około 400, czyli dużo więcej niż liczba obiektów znajdujących się na zdjęciach. Mimo tego regiony należące do różnych klas zostały połączone (prawy górny róg obrazu *sufit_2* oraz tło obrazu *kwiatek_3*).

kwiatek_3



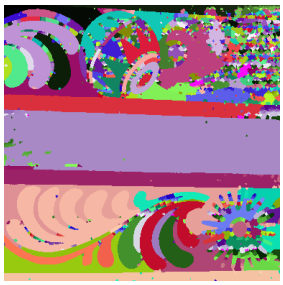
(a)
filtr rozmywający, próg 5% oraz poziom: 10%: *1965* zlewisk

(b)

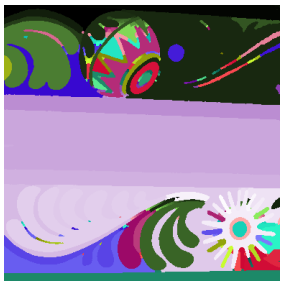
(c)

filtr rozmywający, próg 5% oraz poziom 25%: *410* zlewisk (zlewiska oraz średnie zlewisk)

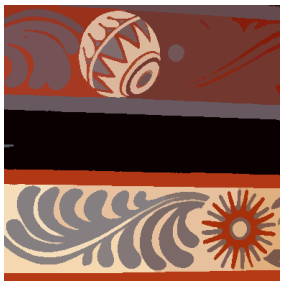
sufit_2



(d)
filtr rozmywający, próg 5% oraz poziom: 10%: *2144* zlewiska



(e)



(f)

filtr rozmywający, próg 5% oraz poziom 20%: *437* zlewisk (zlewiska oraz średnie zlewisk)

Rysunek 3.6: Wpływ łączenia płytkich zlewisk na nadsegmentację. Obrazy (b) i (c) oraz (e) i (f) przedstawiają skutki zastosowania zbyt wysokiego poziomu łączenia.

Powyższe przykłady pokazują, że opisane metody nie pozwalają na satysfakcjonującą redukcję nadsegmentacji bez pogorszenia jakości segmentacji (łączenia różnych klas). Poniżej zostanie zaproponowane rozwiązanie tego problemu.

3.3 Redukcja nadsegmentacji z zastosowaniem analizy skupień

Metody omówione w punkcie 2.7 są często projektowane z myślą o rozwiązaniu konkretnego zadania – przykładem może być problem segmentacji pasów drogi [13, 17]. Projektowane są także uniwersalne metody segmentacji obrazów [33, 43]. Metody należące do pierwszej kategorii trudno dostosować do innych zadań, podczas gdy metody uniwersalne nie zawsze pozwalają na uzyskanie

dobrych wyników. Dlatego autor niniejszej pracy zaproponował [104], aby wykorzystać hierarchiczne metody analizy skupień do łączenia zlewisk w nadmiernie posegmentowanych obrazach. Wybrane zostały ze względu na ich uniwersalność. Można je uważać za modułowe, ponieważ łatwo zmienić metodę klasteryzacji, współczynnik podobieństwa czy atrybuty zlewisk bez modyfikowania pozostałych elementów. Jeżeli metoda zostanie właściwie zaimplementowana wymiana poszczególnych elementów jest bardzo prosta i szybka. Dzięki temu uniwersalną metodę można dostosować do konkretnego celu segmentacji. Gdy pojawi się taka potrzeba możliwe jest także łatwe dodanie nowych metod klasteryzacji, współczynników i atrybutów, co dodatkowo zwiększa możliwości adaptacji proponowanego podejścia. Początkowo testowano je na monochromatycznych obrazach CT oraz MRI [104]. Następnym krokiem było testowanie proponowanej metody na obrazach barwnych [107].

Wykorzystane metody analizy skupień zostały szczegółowo opisane w dodatku A. Redukcja nadsegmentacji z ich wykorzystaniem składa się z trzech etapów:

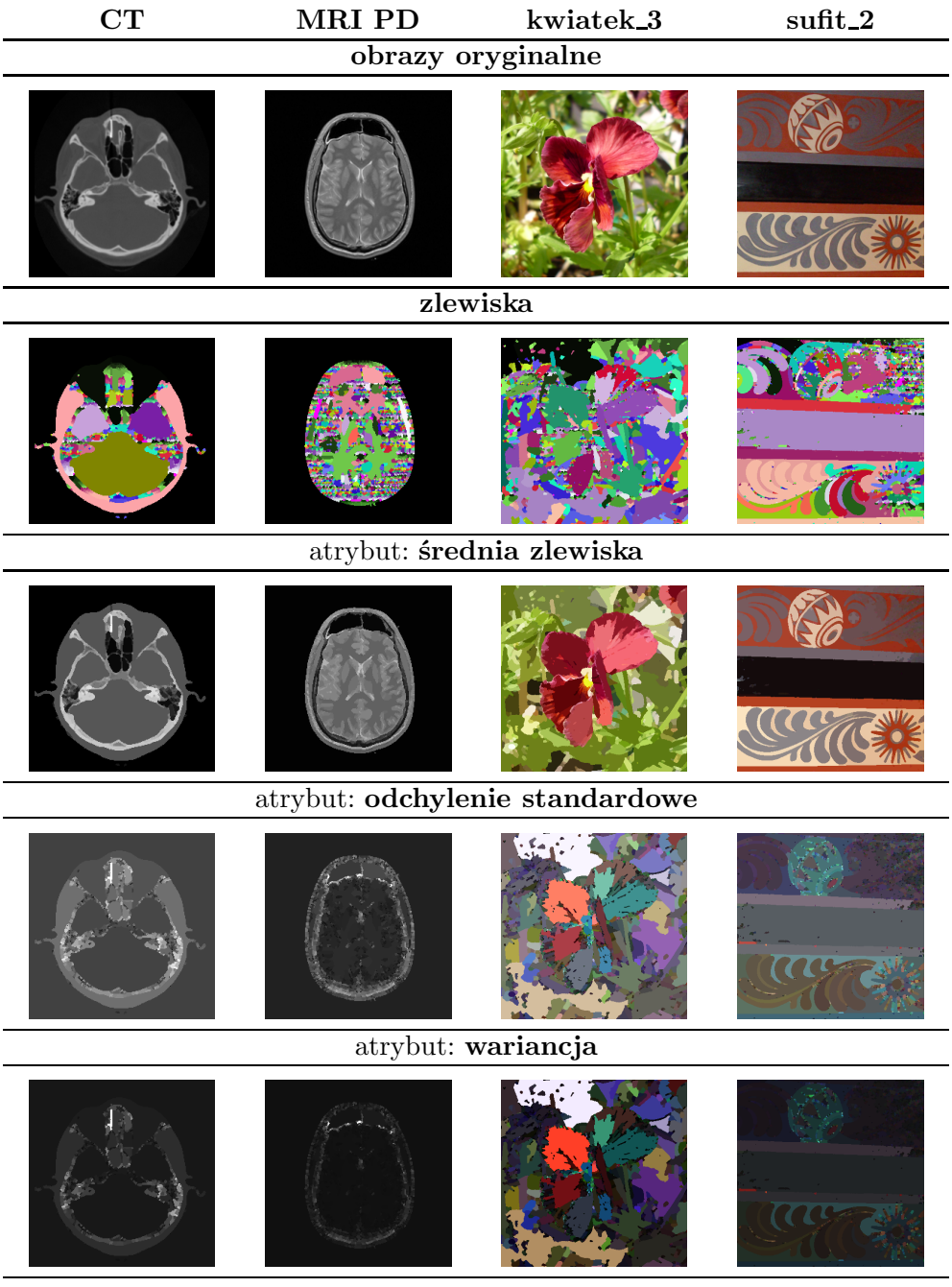
1. Wybór i obliczenie wartości atrybutów wszystkich zlewisk występujących w obrazie. Wykorzystywane rodzaje atrybutów przedstawiono w punkcie 3.3.1.

2. Przeprowadzenie analizy skupień na zlewiskach, które są obiektami opisanymi za pomocą wektorów z atrybutami (punkt 3.3.2).

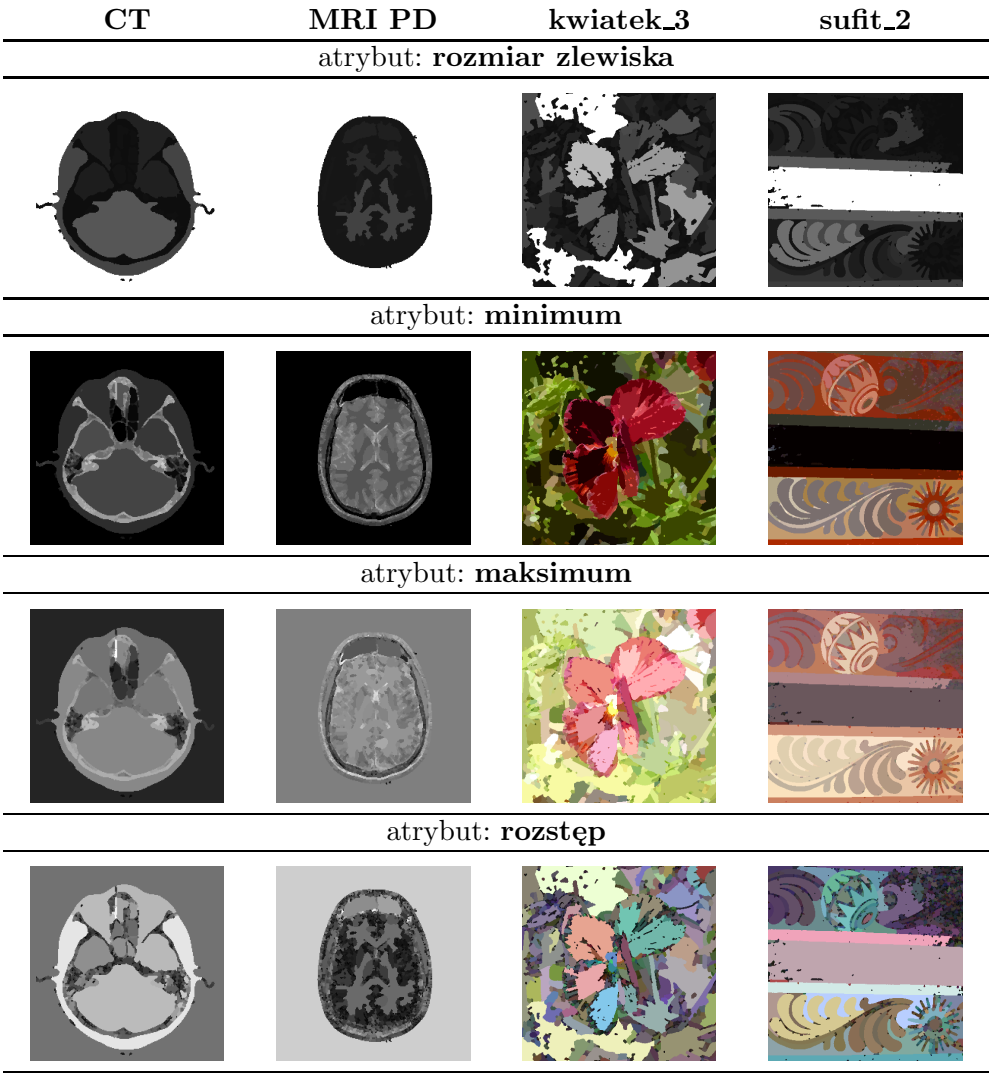
3. Wygenerowanie *tablic LUT*, czyli *tablic podglądu* (ang. *look-up tables*) umożliwiających szybkie łączenie zlewisk w określoną liczbę klas (punkt 3.3.3).

3.3.1 Atrybuty zlewisk

Pierwszym spośród zastosowanych atrybutów jest *rozmiar zlewiska*. Jest on wyrażony liczbą pikseli wchodzących w skład zlewiska. Ten atrybut może być szczególnie przydatny, w sytuacji gdy na przykład jedna lub dwie klasy są nadmiernie posegmentowane. Przykład takiego zjawiska można zaobserwować na rysunku 3.7 (zlewiska), na którym przedstawiono poszczególne zlewiska w zestawieniu z rysunkiem 3.8 (atrybut: rozmiar zlewiska), na którym czarne obszary odpowiadają grupom małych zlewisk. Jak widać opisana sytuacja występuje na obrazach medycznych. W obrazie *CT* tkanka kostna składa się z wielu małych zlewisk, natomiast w obrazie *MRI PD* nadmierna segmentacja występuje w zewnętrznej części głowy, a także w zewnętrznych obszarach mózgu. W przypadku zdjęć wykonanych typowym cyfrowym aparatem fotograficznym obszary o ciemnych barwach są bardzo często mocniej zaszumione, a co za tym idzie zbyt mocno posegmentowane. Zjawisko takie jest widoczne w przypadku obrazu *sufit_2*, w którym szczególnie prawy górny róg jest silnie zaszumiony. W takiej sytuacji rozmiar zlewiska może być przydatny przy segmentacji tych obszarów, co można wykorzystać do selektywnego odszumiania obrazu.



Rysunek 3.7: Ilustracja atrybutów zlewisk zastosowanych w pracy. Atrybuty obliczono dla dwóch monochromatycznych obrazów medycznych oraz dwóch obrazów barwnych.



Rysunek 3.8: Ilustracja atrybutów zlewisk zastosowanych w pracy – ciąg dalszy. Atrybuty obliczono dla dwóch monochromatycznych obrazów medycznych oraz dwóch obrazów barwnych.

Drugi rodzaj atrybutu – *średnia zlewiska* – jest naturalnym i powszechnie stosowanym [24, 31, 43, 52, 97] sposobem opisu zlewiska. Jak widać na rysunku 3.7 segmentacje, w których etykiety zlewisk zastąpiono ich wartością średnią bardzo dobrze przypominają oryginał. Średnia obszaru jest również stosowana w wielu funkcjach oceny segmentacji (między innymi w równaniach 4.11, 4.12, 4.13, 4.16) i służy określeniu stopnia w jakim segmentacja odzwierciedla obraz oryginalny. Wadą tego atrybutu jest to, że nie uzwględnia stopnia zróżnicowania wartości pikseli, co oznacza, że znacznie różniące się obszary mogą mieć taką samą średnią i w celu ich rozróżnienia konieczne jest zastosowanie dodatkowego atrybutu.

Kolejnymi atrybutami są *wariancja zlewiska* oraz *odchylenie standardowe zlewiska*. Są one do pewnego stopnia wrażliwe na teksturę występującą w regionie i mogą stanowić bardzo dobre uzupełnienie średniej. Obszary o bardzo zbliżonych wartościach średnich mogą zauważalnie różnić się pod względem wariancji czy odchylenia standardowego. Dobrymi przykładami takiej sytuacji są obrazy *CT* oraz *kwiatek_3* na rysunku 3.7. W przypadku obrazu *CT* szare obszary wewnątrz i na zewnątrz czaszki wydają się identyczne pod względem średniej, jednak ich wariancja i odchylenie standardowe są znacząco inne. Jeszcze większe różnice widoczne są dla obrazu *kwiatek_3*. Dość podobne pod względem barwy bordowe płatki kwiatka, a także zielone liście różnią się znacząco pod względem wariancji, co widać na obrazach przedstawiających wartości wariancji czy odchylenia standardowego zlewiska. Dodatkowo można zauważyć, że zlewiska różnią się nie tylko pod względem jasności ale też barwy. Oznacza to, ich powierzchnie, odpowiadające podobnym zielonym liściom czy bordowym płatkom, charakteryzują się różną zmiennością w poszczególnych barwnych składowych.

Miarą określającą zakres, do którego należą wartości wewnątrz poszczególnych zlewisk jest kolejny atrybut zlewiska czyli *rozstęp*. Jest on również częściowo zależny od tekstury poszczególnych obszarów, i co można zauważyć na rysunkach 3.7 oraz 3.8, istnieje pewna zależność pomiędzy wartościami rozstępu i odchylenia standardowego. Rozstęp bardziej wyolbrzymia różnice między wartościami, ponieważ nie uwzględnia liczby punktów o określonej wartości.

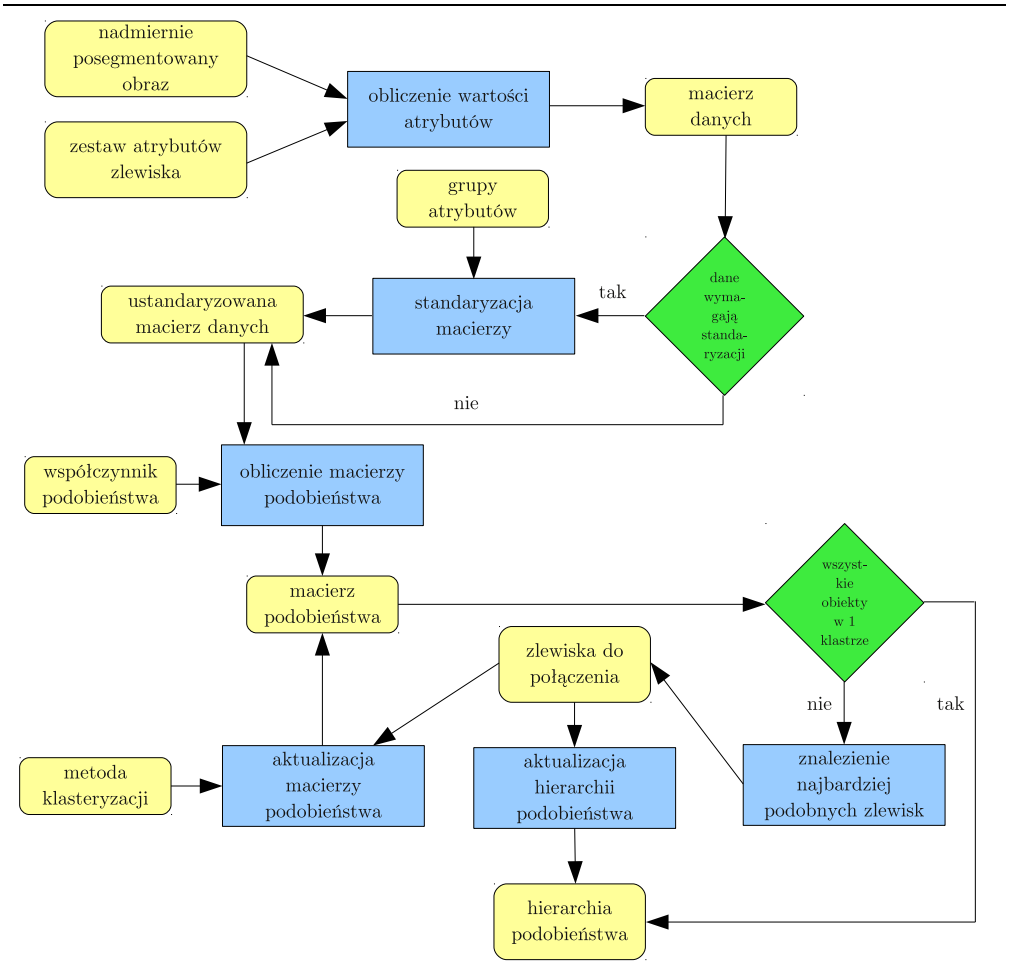
Minimalna oraz *maksymalna wartość zlewiska* są atrybutami pozwalającymi na różnicowanie zlewisk na podstawie niedokładności transformacji wododziałowej lub uniezależnienie się od jej niedokładności. Dobrym przykładem mogą być wartości tych atrybutów uzyskane dla obrazu *MRI PD*. Przedstawiono je na rysunku 3.8. W przypadku maksymalnej wartości zlewiska wyraźnie widać, że w tym samym zlewisku, co czarne tło znalazła się także część jaśniejszych punktów, należących do innego obiektu. Widoczne jest również, że w wypadku tego obrazu minimalne wartości lepiej charakteryzują poszczególne zlewiska. Tak więc, zastosowanie wartości minimalnej (maksymalnej) pozwala na uniezależnienie wyników segmentacji od błędów transformacji wododziałowej – na przykład gdy do ciemnego (jasnego) zlewiska zostanie włączona pewna liczba jasnych (ciemnych) punk-

tów. Atrybuty te razem z rozstępem pozwalają na wykorzystanie wspomnianych niedokładności w celu rozróżniania zlewisk. Przykładem takiej sytuacji może być przypadek, w którym zlewiska leżące na brzegu nadmiernie posegmentowanego obiektu będą zawierały piksele barwą/jasnością lepiej odpowiadającą sąsiadom (co jest dość prawdopodobne), podczas gdy zlewiska leżące wewnątrz obiektu nie będą zawierały takich pikseli. Wtedy rozstęp oraz minimum lub maksimum (w zależności od jasności obiektu) pozwalają na rozróżnienie zlewisk leżących na brzegu, od tych, leżących we wnętrzu obiektu.

Atrybuty obliczane są na podstawie obrazu oryginalnego oraz obrazu z etykietami zlewisk (z wyjątkiem rozmiaru zlewiska obliczanego wyłącznie na podstawie etykiet). W przypadku obrazów w skali szarości każdy atrybut to jedna wartość. W przypadku obrazów barwnych wyznaczenie wartości jednego atrybutu takiego jak średnia oznacza uzyskanie dla jednego zlewiska trzech wartości, po jednej dla poszczególnych składowych. Oczywiście nic nie stoi na przeszkodzie, aby obliczać wartości atrybutów tylko dla wybranych składowych. Obliczanie kilku wartości dla jednego rodzaju atrybutu może wymagać modyfikacji sposobu standaryzacji i grupowania tych wartości. Bezpośrednie zastosowanie równań A.21 – A.24 mogłoby prowadzić do zniekształcenia informacji o barwie.

3.3.2 Klasteryzacja zlewisk

Przebieg klasteryzacji zlewisk pokazano schematycznie na rysunku 3.9. Polega ona na potraktowaniu zlewisk w obrazie jak obiektów opisanych za pomocą zestawu atrybutów ilościowych. Kolejnym, opcjonalnym, krokiem jest standaryzacja umożliwiająca wyrównanie wpływu poszczególnych atrybutów na wartości współczynnika podobieństwa. Dostępne sposoby standaryzacji przedstawiono w części A.5 dodatku. Stopień podobieństwa zlewisk jest mierzony za pomocą jednego z siedmiu współczynników podobieństwa lub odległości opisanych w punkcie A.4. Współczynnik wykorzystywany jest do obliczenia macierzy podobieństwa, w której zapisywane są jego wartości dla wszystkich możliwych par obiektów. Następnie przeprowadzany jest proces klasteryzacji zlewisk, w trakcie którego są one łączone ze sobą w grupy. Wszystkie połączenia zapamiętywane są w hierarchii podobieństwa (podobnej do tej przedstawionej na rysunku A.1), a wartości podobieństwa lub odległości dla klastrów powstających w trakcie tego procesu, obliczane są za pomocą jednej z metod przedstawionych w punkcie A.3.



Rysunek 3.9: Schemat ilustrujący zastosowanie metod analizy skupień do obliczenia hierarchii podobieństwa zlewisk.

Tabela 3.2: Zestawienie dodatkowych oznaczeń użytych w algorytmie generującym hierarchię podobieństwa zlewisk

Metoda/pole	**Opis**
obiekt: atrybutyZlewisk – lista zawierająca wektory atrybutów wszystkich zlewisk	
`liczbaZlewisk`	Zawiera liczbę zlewisk w nadmiernie posegmentowanym obrazie.

ciąg dalszy na następnej stronie...

Tabela 3.2 – ciąg dalszy

Metoda/pole	**Opis**
`obliczWartosci(` *`lista, skladowe, obrazWe, obrazWy`*`)`	Oblicza i zapamiętuje w obiekcie wartości atrybutów dla wszystkich zlewisk w obrazie. Parametr *lista* zawiera nazwy atrybutów. Parametr *skladowe* określa, dla której składowej obrazu obliczyć odpowiadający mu atrybut z listy. Parametry *obrazWe* i *obrazWy* zawierają odpowiednio obraz oryginalny i obraz z etykietami.
`standaryzuj(` *`metodaStandaryzacji, grupyAtrybutow`*`)`	Używając metody standaryzacji, przekazanej jako pierwszy parametr, modyfikuje zapisane na liście wektory atrybutów zlewisk. Parametr *grupyAtrybutow* określa, które atrybuty mają być standaryzowane razem.
obiekt: macierzPodobienstwa – przechowuje wartości podobieństwa określone dla wszystkich par klastrów, które można połączyć.	
`obliczPodobienstwo(` *`atrybuty, wspolczynnik`*`)`	Na podstawie listy z wektorami atrybutów zlewisk i współczynnika podobieństwa oblicza wartość miary podobieństwa dla wszystkich możliwych par zlewisk. Metoda służy do obliczenia wartości początkowych.
`polaczKlastry(` *`metodaKlasteryzacji, skladKlastrow, klaster1, klaster2`*`)`	Łączy klastry o podanych numerach. Nowy klaster ma numer *klaster1*. Używając zadanej metody klasteryzacji, oblicza podobieństwo nowego klastra do wszystkich pozostałych. Do obliczeń konieczna jest informacja o składzie poszczególnych klastrów.
`usunKlaster(klaster)`	Usuwa klaster o zadanym numerze.
`znajdzPodobneKlastry(wspolczynnik)`	Funkcja znajduje nowych kandydatów do połączenia i zwraca obiekt *polaczenieZlewisk*, zawierający informację o numerach klastrów do połączenia oraz wartości miary podobieństwa. W zależności od zastosowanego współczynnika funkcja szuka klastrów o maksymalnym podobieństwie lub minimalnej różnicy.
obiekt: skladKlastrow – przechowuje etykiety zlewisk wchodzące w skład klastrów o określonych numerach.	
`ustawPoczatkowy(` *`liczbaZlewisk`*`)`	Ustawia początkowy skład klastrów. Początkowo każdy klaster składa się z jednego zlewiska o takim samym numerze.
`liczbaKlastrow`	Zawiera bieżącą liczbę klastrów.

ciąg dalszy na następnej stronie...

Tabela 3.2 – ciąg dalszy

Metoda/pole	**Opis**
`polaczKlastry(` *`klaster1, klaster2`*`)`	Łączy dwie listy etykiet. Nowa lista ma numer *klaster1*.
`usunKlaster(` *`klaster`*`)`	Usuwa spis zlewisk klastra o zadanym numerze.
`reprezentant(` *`klaster`*`)`	Zwraca numer zlewiska, które jest bieżącym reprezentantem klastra. Reprezentantem jest zlewisko o najniższym numerze wchodzące w skład klastra.
obiekt: polaczenia – lista przechowująca historię połączeń klastrów. Połączenia pamiętane są w kolejności dodawania. Lista umożliwia wygenerowanie segmentacji z połączonymi zlewiskami.	
`dodajKolejne(` *`reprezentant1, reprezentant2`*`)`	Dodaje do listy informację o połączeniu dwóch klastrów. Zapisywana jest informacja o reprezentantach łączonych klastrów.
`liczbaPolaczen`	Zawiera liczbę połączeń klastrów.
`operator[`*`numer`*`]`	Zwraca zapisany na liście obiekt opisujący połączenie dwóch klastrów.
klasa: Polaczenie – reprezentuje jednostkowe połączenie klastrów. Zawiera informacje potrzebne do łączenia zlewisk.	
`reprezentant1`	Zawiera numer reprezentanta pierwszego z łączonych klastrów. Reprezentantem jest zlewisko o najniższym numerze w klastrze.
`reprezentant2`	Zawiera numer reprezentanta drugiego z łączonych klastrów. Reprezentantem jest zlewisko o najniższym numerze w klastrze.
obiekt: polaczenieKlastrow – roboczy obiekty reprezentujący jednostkowe połączenie klastrów. Wykorzystywany w trakcie tworzenia hierarchii podobieństwa.	
`pierwszy`	Zawiera numer pierwszego z łączonych klastrów.
`drugi`	Zawiera numer drugiego z łączonych klastrów.
`podobienstwo`	Zawiera wartość miary podobieństwa łączonych klastrów.

Szczegółowy opis procesu klasteryzacji zlewisk przedstawia algorytm 3.4. Pierwszym krokiem jest obliczenie listy *atrybutyZlewisk* zawierającej wektory z wartościami atrybutów dla poszczególnych zlewisk. Wartości atrybutów obliczane są na podstawie obrazu oryginalnego reprezentowanego przez obiekt *obrazWe* oraz obrazu nadmiernie posegmentowanego zawierającego etykiety zlewisk – *obrazWy*. Ponadto konieczna jest *listaAtrybutow* zawierająca opis zestawu atrybutów. A w przypadku obrazów barwnych konieczna jest również lista *skladowe*

określająca, dla których składowych obrazu należy obliczyć poszczególne atrybuty. Dzięki temu w procesie grupowania zlewisk możliwe jest uwzględnienie tylko wybranych składowych obrazu. W dalszej kolejności może być zastosowana standaryzacja wartości atrybutów. Jej przeprowadzenie wymaga wybrania jednej z czterech dostępnych metod oraz określenia grup atrybutów. Wprowadzenie grupowania atrybutów pozwala na zachowanie proporcji między wartościami tego samego atrybutu obliczonymi dla różnych składowych. Dzięki temu standaryzacja nie zniekształca informacji o barwie. Kolejnym krokiem jest wyznaczenie macierzy podobieństwa. Wartości zapisane w macierzy obliczane są dla wszystkich możliwych par zlewisk przy użyciu wybranego współczynnika podobieństwa. Po obliczeniu macierzy, w obiekcie *skladKlastrow* zapisywany jest początkowy skład klastrów. Liczba klastrów odpowiada liczbie zlewisk, zatem każdy klaster zawiera tylko jedno zlewisko. Po ustawieniu wartości początkowych algorytm rozpoczyna budowanie hierarchii podobieństwa. Kolejne iteracje wykonywane są do momentu, w którym wszystkie zlewiska znajdą się w jednym klastrze. Na początku każdej iteracji algorytm znajduje w macierzy podobieństwa dwa najbardziej podobne do siebie klastry. Znajdowani są reprezentanci łączonych klastrów. Reprezentantem jest zlewisko o najmniejszym numerze wchodzące w skład klastra. W każdej iteracji do listy *polaczenia* przechowującej historię połączeń dodawane są połączenia klastrów przechowywane jako uporządkowane pary, w których pierwszy element jest mniejszy od drugiego. Po zapisaniu połączenia uaktualniana jest macierz podobieństwa. Przy wykorzystaniu określonej metody klasteryzacji obliczane są wartości miary podobieństwa/odległości między nowym a pozostałymi klastrami. Połączony klaster dziedziczy numer pierwszego z łączonej pary. Drugi klaster jest kasowany. Po macierzy podobieństwa uaktualniane są składy klastrów. Łączone są listy zlewisk składowych łączonych klastrów a także zmniejszana jest aktualna liczba klastrów przechowywanych na liście.

Po zakończeniu tego etapu istnieje historia kolejnych połączeń klastrów przechowywanych w obiekcie *polaczenia*. Historia połączeń jest postacią hierarchii podobieństwa umożliwiającą szybkie łączenie zlewisk w klasy w następnym etapie algorytmu.

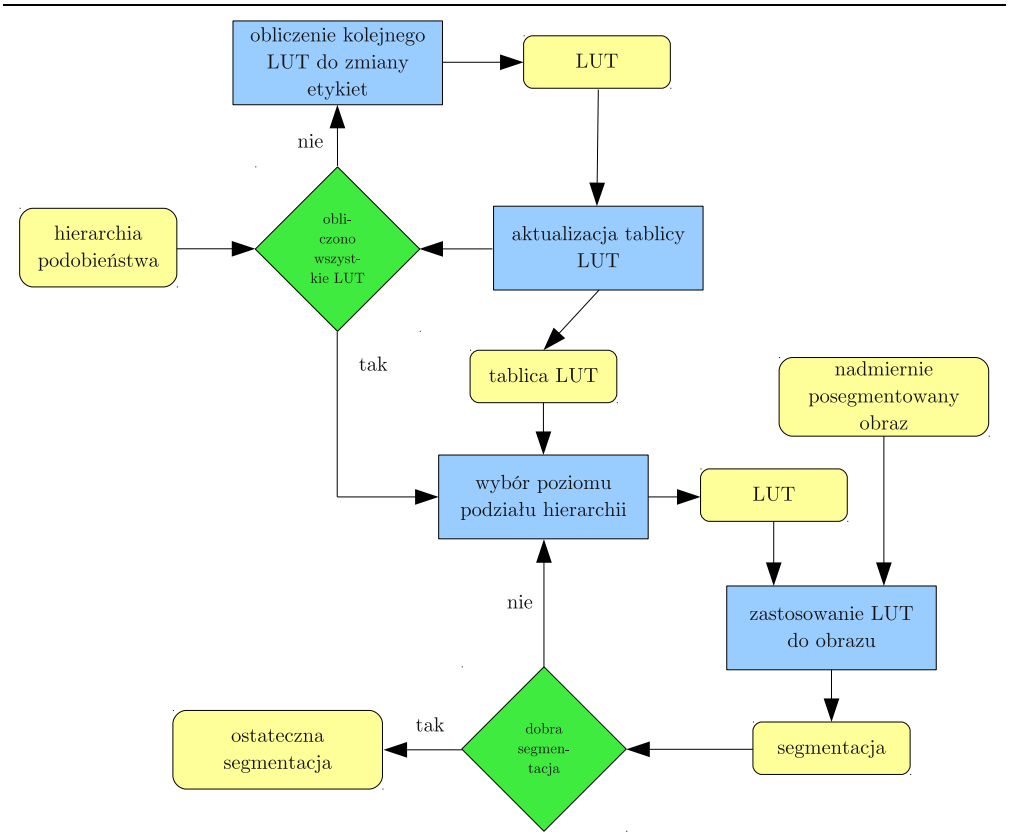
Mimo, że algorytm 3.4 został sformułowany z myślą o metodach pojedynczego (SLINK), pełnego (CLINK) i średniego wiązania (UPGMA) opisanych w dodatku A, to może on też posłużyć do przeprowadzenia klasteryzacji z użyciem metody Warda. Wtedy jako współczynnik podobieństwa należy przyjąć wzór A.10 i wypełnić macierz podobieństwa wartościami błędu kwadratowego obliczonymi dla wszystkich możliwych klastrów powstałych z dwóch zlewisk (na początku klastry składają się z pojedynczych zlewisk). Aktualizacja macierzy (wykonywana przez metodę `polaczKlastry` obiektu *macierzPodobienstwa*) polega na obliczeniu sumy-kwadratów E_p (wzór A.10) dla wszystkich możliwości połączeń nowego klastra z pozostałymi. Największych zmian wymaga poszukiwanie kandydatów

Algorytm 3.4 Algorytm klasteryzacji zlewisk

```
1: atrybutyZlewisk.obliczWartosci(listaAtrybutow,skladowe,obrazWe,
   obrazWy)
2: if standaryzowac = true then
3:   atrybutyZlewisk.standaryzuj(metodaStandaryzacji,grupyAtrybutow)
4: end if
5: macierzPodobienstwa.obliczPodobienstwo(atrybutyZlewisk,
   wspolczynnikPodobientwa)
6: skladKlastrow.ustawPoczatkowy(atrybutyZlewisk.liczbaZlewisk)
7: while skladKlastrow.liczbaKlastrow > 1 do
8:   polaczenieKlastrow ←
     macierzPodobienstwa.znajdzPodobneKlastry(wspolczynnik)
9:   reprezentant1 ←
     skladKlastrow.reprezentant(polaczenieKlastrow.pierwszy)
10:  reprezentant2 ←
     skladKlastrow.reprezentant(polaczenieKlastrow.drugi)
11:  if reprezentant2 < reprezentant1 then
12:    robocza ← reprezentant2
13:    reprezentant2 ← reprezentant1
14:    reprezentant1 ← robocza
15:  end if
16:  polaczenia.dodajKolejne(reprezentant1, reprezentant2)
17:  macierzPodobienstwa.polaczKlastry(metodaKlasteryzacji,
     skladyKlastrow, polaczenieKlastrow.pierwszy,
     polaczenieKlastrow.drugi)
18:  macierzPodobienstwa.usunKlaster(polaczenieKlastrow.drugi)
19:  skladKlastrow.polaczKlastry(polaczenieKlastrow.pierwszy,
     polaczenieKlastrow.drugi)
20:  skladKlastrow.usunKlaster(polaczenieKlastrow.drugi)
21: end while
```

do połączenia. W obiekcie *macierzPodobienstwa* przechowywana jest również tablica pomocnicza, która zawiera błędy E_p poszczególnych klastrów. Podczas wyszukiwania sumowane są wszystkie elementy tej tablicy. Otrzymywany jest w ten sposób sumaryczny błąd E (wzór A.9) wszystkich klastrów utworzonych do tej pory. Następnie przy użyciu macierzy podobieństwa znajdowane są takie klastry i i j, że nowa wariancja $E' = E - E_i - E_j + E_{ij}$ jest najmniejsza (E_{ij} to wariancja nowego klastra powstałego po połączeniu i i j).

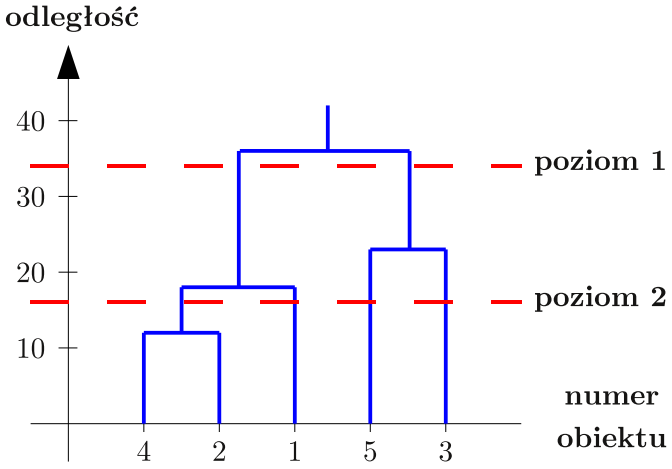
Warto również dodać, że szybszy i prostszy sposób, polegający na przechowywaniu w jednej zmiennej sumy wariancji klastrów utworzonych w poprzednich iteracjach, a także odpowiednie korygowanie jej podczas szukania kandydatów do połączenia, nie daje dobrych efektów. Przy typowej liczbie zlewisk, występujących w nadmiernie posegmentowanych obrazach, takie postępowanie prowadzi do bardzo dużych błędów obliczeń numerycznych.



Rysunek 3.10: Schemat przedstawiający sposób uzyskania ostatecznej segmentacji na podstawie hierarchii podobieństwa oraz nadmiernie posegmentowanego obrazu.

3.3.3 Uzyskanie ostatecznej segmentacji

Ostatnim krokiem redukcji nadsegmentacji z wykorzystaniem analizy skupień jest faktyczne połączenie zlewisk. Proces łączenia został przedstawiony schematycznie na rysunku 3.10. Polega on na utworzeniu serii tablic LUT czyli tablic podglądu (ang. *look-up tables*). Pierwsza tablica LUT odpowiada wykonaniu tylko jednego połączenia klastrów, druga – dwóch itd. Każda z tablic zawiera etykiety, jakie należy nadać poszczególnym zlewiskom w ostatecznej segmentacji – zlewiska są łączone poprzez nadanie im jednolitych etykiet. Tworzenie pojedynczej tablicy LUT nie jest długotrwałe, a więc tworzonych jest tyle tablic, ile jest zarejestrowanych połączeń klastrów. Dostępność wszystkich tablic umożliwia z kolei, szybkie generowanie segmentacji z określoną liczbą klas i interaktywny wybór tej liczby. Wybór konkretnej liczby klas (i tablicy LUT) jest równoważny z przecięciem hierarchii podobieństwa, przedstawionej jako drzewo, na określonym poziomie. Rysunek 3.11 przedstawia prostą hierarchię, na której zaznaczono dwa poziomy. Wybranie *poziomu 1* jest równoważne z wykonaniem wszystkich połączeń



Rysunek 3.11: Podział hierarchii podobieństwa umożliwia uzyskanie określonej liczby klas. Na podstawie tej samej hierarchii można wygenerować wiele różnych segmentacji.

w części hierarchii podobieństwa znajdującej się poniżej tego poziomu i oznacza powstanie dwóch klas: (1;2;4) oraz (3;5). Wybranie *poziomu 2* powoduje uzyskanie czterech klas: (1), (2), (4) oraz (3;5). Tworzenie segmentacji jest proste i wymaga zastosowania wybranej tablicy do obrazu zawierającego etykiety zlewisk. Algorytm przegląda obraz ze zlewiskami i zamienia kolejne wartości punktów na wartości zapisane w wybranej tablicy LUT.

Tabela 3.3: Zestawienie dodatkowych oznaczeń użytych w algorytmie generującym tablice LUT

Metoda/pole	**Opis**
obiekt: LUTy – tablica złożona z obiektów klasy LUT. Przechowuje tablice LUT umożliwiające szybkie generowanie segmentacji zawierających wybraną liczbę klas.	
`operator[`*`liczba`*`]`	Zwraca tablicę LUT, która odpowiada *liczba+1* połączeń zlewisk.
obiekt: roboczyLUT – tymczasowy obiekt klasy LUT. Wykorzystywany w trakcie budowania tablicy.	
klasa: LUT – obiekty tej klasy przechowują etykiety poszczególnych zlewisk w ostatecznej segmentacji. Nadanie zlewiskom tej samej etykiety oznacza ich połączenie.	
`operator[`*`zlewisko`*`]`	Zwraca etykietę zlewiska o zadanym numerze.

ciąg dalszy na następnej stronie...

Tabela 3.3 – ciąg dalszy

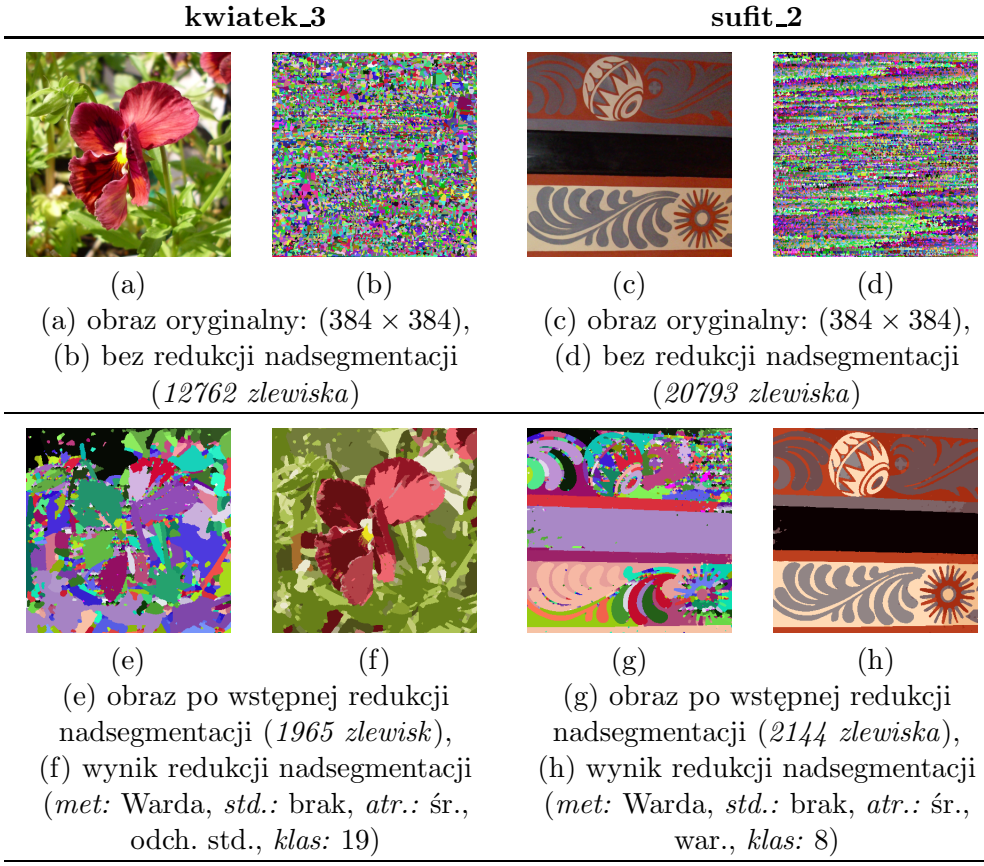
Metoda/pole	Opis
`zeruj()`	Zeruje wszystkie elementy tablicy.

Algorytm 3.5 prezentuje sposób tworzenia serii tablic LUT na podstawie historii połączeń klastrów. Nowe oznaczenia wykorzystane wyjaśniono w tabeli 3.3. W kolejnych iteracjach tworzone są kolejne tablice LUT odpowiadające wykonaniu jednego połączenia klastrów więcej.

Algorytm 3.5 Algorytm tworzenia tablic LUT umożliwiających nadawanie zlewiskom wspólnych etykiet

```
1: nrPolaczenia ← polaczenia.liczbaPolaczen − 1
2: while nrPolaczenia ⩾ 0 do
3:   roboczyLUT.zeruj()
4:   nastepnaEtykieta ← 0
5:   nrPolaczeniaRob ← nrPolaczenia
6:   while nrPolaczeniaRob ⩾ 0 do
7:     reprezentant1 ← polaczenia[nrPolaczenia].reprezentant1
8:     reprezentant2 ← polaczenia[nrPolaczenia].reprezentant2
9:     if roboczyLUT[reprezentant1] > 0 then
10:      roboczyLUT[reprezentant2] ← roboczyLUT[reprezentant1]
11:    else
12:      nastepnaEtykieta ← nastepnaEtykieta + 1
13:      roboczyLUT[reprezentant1] ← nastepnaEtykieta
14:      roboczyLUT[reprezentant2] ← nastepnaEtykieta
15:    end if
16:    nrOdlegosciRob ← nrPolaczeniaRob − 1
17:  end while
18:  for nrZlewiskaRob = 0 to atrybutyZlewisk.liczbaZlewisk do
19:    if roboczyLUT[nrZlewiskaRob] = 0 then
20:      nastepnaEtykieta ← nastepnaEtykieta + 1
21:      roboczyLUT[nrZlewiskaRob] ← nastepnaEtykieta
22:    end if
23:  end for
24:  LUTy[nrPolaczenia] ← roboczyLUT
25:  nrPolaczenia ← nrPolaczenia − 1
26: end while
```

Utworzenie konkretnej tablicy LUT rozpoczyna się od jej wyzerowania. Przyjęto, iż zero w tablicy LUT oznacza, że dla danego zlewiska nie określono jeszcze etykiety w końcowej segmentacji. Następnie nadawane są etykiety zlewiskom, które będą połączone. Algorytm przemieszcza się w dół hierarchii podobieństwa – od połączenia klastrów znajdującego się najwyżej w tej hierarchii do połączenia znajdującego się najniżej. Oczywiście pod uwagę brana jest tylko dolna część



(a) (b) (c) (d)

(a) obraz oryginalny: (384 × 384), (b) bez redukcji nadsegmentacji (*12762 zlewiska*)

(c) obraz oryginalny: (384 × 384), (d) bez redukcji nadsegmentacji (*20793 zlewiska*)

(e) (f) (g) (h)

(e) obraz po wstępnej redukcji nadsegmentacji (*1965 zlewisk*), (f) wynik redukcji nadsegmentacji (*met:* Warda, *std.:* brak, *atr.:* śr., odch. std., *klas:* 19)

(g) obraz po wstępnej redukcji nadsegmentacji (*2144 zlewiska*), (h) wynik redukcji nadsegmentacji (*met:* Warda, *std.:* brak, *atr.:* śr., war., *klas:* 8)

Rysunek 3.12: Wyniki redukcji nadsegmentacji wykorzystującej analizę skupień.

tej hierarchii znajdująca się poniżej wybranego progu, który określa liczbę klas w ostatecznej segmentacji. Algorytm odczytuje połączenie i zapisanych w nim reprezentantów łączonych klastrów. Jeżeli pierwszy z nich posiada już etykietę (różną od zera), to jest ona kopiowana do drugiego z reprezentantów. Jeżeli pierwszy z nich nie posiada etykiety, to obaj reprezentanci (oba zlewiska) otrzymują nową unikalną etykietę. Następnie algorytm przechodzi do kolejnego połączenia. Przyjęty sposób wyboru reprezentantów klastrów, uporządkowanie reprezentantów na liście połączeń oraz kierunek odczytywania połączeń powoduje, że zlewiska wchodzące w skład poszczególnych klastrów etykietowane są w kolejności, od tego o najniższym do tego o najwyższym numerze. Po nadaniu etykiet łączonym zlewiskom algorytm uzupełnia tworzoną tablicę LUT etykietami zlewisk, które nie zostaną połączone. Wszystkim zlewiskom posiadającym etykietę zero nadawane są kolejne unikalne etykiety.

Na rysunku 3.12 przedstawiono wyniki redukcji nadsegmentacji z wykorzystaniem zaproponowanej metody. Zlewiska były łączone w nadmiernie posegmentowanych obrazach uzyskanych w wyniku wstępnej redukcji nadsegmentacji. Obrazy przygotowano, tak jak opisano w punkcie 3.2. Wykorzystano selektywny filtr,

progowanie obrazu gradientowego oraz zalewanie płytkich zlewisk z parametrami identycznymi do tych w poprzednich przykładach. W szczególności wartość progu oraz poziomu dobrano w taki sposób, aby nie zostały połączone zlewiska wchodzące w skład różnych obiektów.

Jak widać łączenie zlewisk z wykorzystaniem analizy skupień pozwala na znaczną redukcję nadsegmentacji. W przypadku obrazu testowego *kwiatek_3* liczba zlewisk zmniejszyła się z 1965 (po wstępnej redukcji) do 19, a w przypadku obrazu *sufit_2* z 2144 do 8. Nie wystąpiły przy tym wyraźnie widoczne na rysunkach 3.4c, 3.6c, 3.4f oraz 3.6f efekty zastosowania zbyt wysokiego progu oraz poziomu łączenia.

Rozdział 4

Dobór parametrów analizy skupień

4.1 Problem jakości segmentacji

Mimo że problem oceny jakości segmentacji cieszy się w literaturze mniejszym zainteresowaniem niż sama segmentacja, to jednak, powstało wiele opracowań na ten temat. W jednym z najbardziej znanych Haralick i Shapiro sformułowali postulaty, które określają co decyduje o tym, że segmentacja jest dobra [41]:

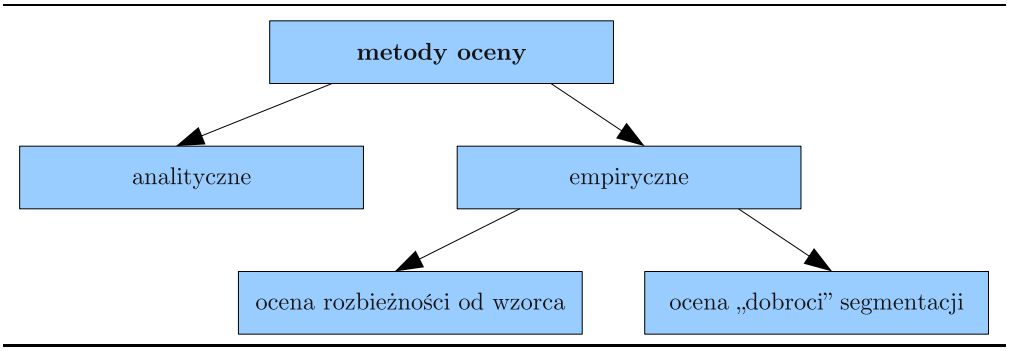
1. Regiony powinny być jednorodne ze względu na określone kryteria. Kryterium takim może być jednolity odcień szarości czy taka sama tekstura.

2. Sąsiadujące ze sobą regiony, powinny znacząco się różnić według tego samego kryterium, według którego każdy z nich jest jednorodny.

3. Wnętrze każdego z regionów powinno być proste i nie powinno posiadać zbyt wielu małych dziur.

4. Krawędzie powinny być gładkie (nie poszarpane) i właściwie zlokalizowane (przykładowo nie powinny być przesunięte względem krawędzi rzeczywistych obiektów).

Praca Haralicka i Shapiro jest głównie obszernym przeglądem technik segmentacji, ale zazwyczaj jest cytowana właśnie ze względu na wymienione postulaty. Stały się one faktycznym standardem, na który powołują się twórcy nowych metod oceny jakości segmentacji. Przykładem mogą być tu Liu i Yanga [62] oraz Borsotti i in. [23]. Ponadto są często wymieniane w pracach przeglądowych dotyczących oceny jakości segmentacji np. [127] czy [128].



Rysunek 4.1: Podział metod oceny jakości segmentacji według Y.J.Zhanga [128].

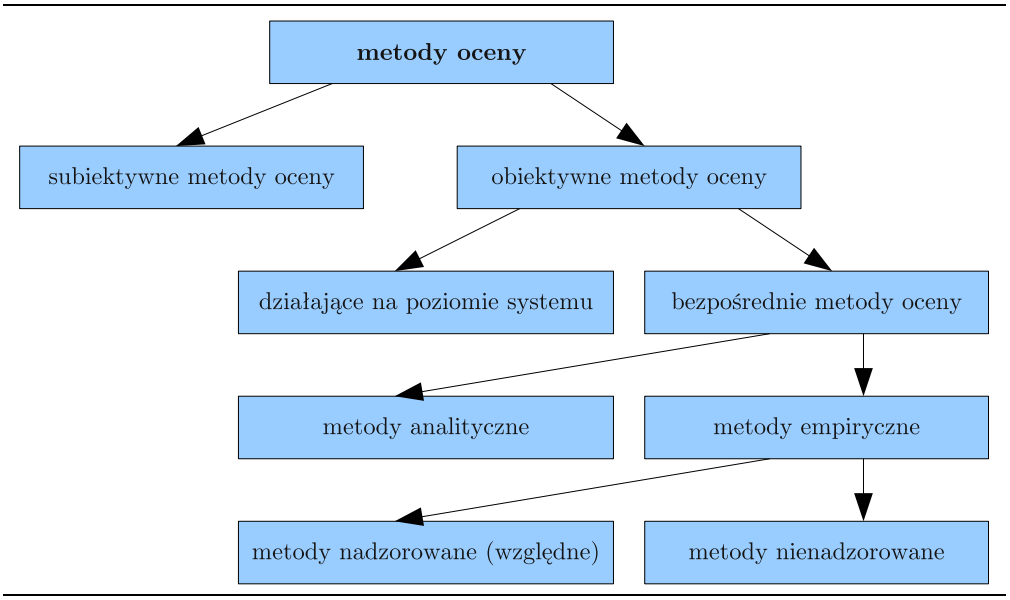
4.2 Podział metod oceny

Y. J. Zhang w swojej pracy [128] zaproponował podział metod oceny algorytmów segmentacji (rysunek 4.1). W pierwszej kolejności wyróżnił metody analityczne oraz empiryczne.

Metody analityczne badają bezpośrednio sam algorytm, a nie wyniki jego działania. Mogą brać pod uwagę zasadę działania algorytmu, jego wymagania czy złożoność obliczeniową. Zaletą metod analitycznych jest to, że są niezależne od implementacji (i potencjalnych błędów) oraz zastosowanych narzędzi (komputer, system operacyjny, narzędzia programistyczne). Analityczne metody oceny segmentacji rzadko pozwalają na uzyskanie ilościowych ocen algorytmu.

Metody empiryczne oceniają algorytm segmentacji na podstawie wyników jego działania. Dzielą się na dwie kategorie, z których pierwsza pozwala odpowiedzieć na pytanie, jak dobry jest wynik segmentacji, a druga mówi jak bardzo wynik segmentacji różni się od zadanego wzorca. Większość metod oceniających "dobroć" segmentacji opiera się na miarach jakości odpowiadających intuicyjnemu rozumieniu pojęcia *dobra segmentacja* przez człowieka. Ich charakterystyczną cechą jest to, że do działania nie wymagają żadnej informacji *a priori*. Z drugiej strony znajdują się metody oceniające rozbieżność między wynikiem segmentacji a zadanym wzorcem. Powoduje to wzrost wymaganego nakładu pracy, ponieważ wymaga przygotowania wzorcowych segmentacji oraz uniemożliwia zastosowanie tych metod w sytuacjach, gdy nie jest możliwe dostarczenie wzorca. Przykładem mogą być systemy czasu rzeczywistego, gdzie na wejście podawane są różnorodne obrazy, których zawartość nie zawsze można przewidzieć.

W nowszym opracowaniu H. Zhang i in. [127] zaproponowali inny podział metod oceny algorytmów segmentacji (rysunek 4.2). Przede wszystkim wyróżnili metody subiektywne oraz obiektywne. Według autorów subiektywny sposób oceniania jest zawsze dokonywany przez ludzi. Oceny przyznawane przez różne osoby mogą się istotnie różnić, co stanowi główną wadę tego podejścia. Ponadto, aby uzyskać znaczące, nieobarczone dużym błędem systematycznym wyniki, należy



Rysunek 4.2: Podział metod oceny jakości segmentacji według H.Zhanga i in. [127].

zaangażować wielu oceniających. W wielu zastosowaniach jest to niepraktyczne.

Metody obiektywne zostały podzielone na metody działające na poziomie pewnego systemu oraz metody bezpośrednio oceniające algorytm segmentacji. W pierwszej grupie metod obiektywnych dokonuje się pośredniej oceny jakości segmentacji, poprzez badanie jej wpływu na system, w ramach którego pracuje sprawdzany algorytm. Niestety, nawet jeśli jakiś system pracuje poprawnie to nie oznacza to, że otrzymuje segmentacje dobrej jakości. Jedyne co można stwierdzić to fakt, iż segmentacje są dostatecznie dobre. Jest to istotna wada takiego podejścia. Druga kategoria - tzn. bezpośrednie metody oceny zostały dalej podzielone na analityczne oraz empiryczne. Metody analityczne, tak jak w klasyfikacji Y. J. Zhanga, oceniają zasadę działania algorytmu, natomiast metody empiryczne wyniki, których on dostarcza. Ta ostatnia grupa została jeszcze podzielona na metody nadzorowane i nienadzorowane. Metody nadzorowane, czasami nazywane również metodami względnymi, polegają na porównaniu segmentacji otrzymanej za pomocą ocenianego algorytmu z wzorcem. Często takim wzorcem jest ręczna segmentacja, którą jej twórca uznał za najlepszą.

Ostatnia kategoria metod – czyli metody nienadzorowane – jest według Zhanga bardzo użyteczna ze względu na to, że stanowią ją metody autonomiczne, często nie wymagające określania żadnych parametrów ani segmentacji wzorcowych. Metody nienadzorowane przyznają segmentacjom oceny ilościowe i obiektywne. Ich zadaniem jest stwierdzenie, które obrazy po segmentacji posiadają cechy pożądane przez ludzi. Dzięki swoim cechom nadają się do zastosowania w automatycznych systemach czasu rzeczywistego [127], a także do sterowania

przebiegiem segmentacji [87]. Do wad nienadzorowanych metod oceny segmentacji [127] można zaliczyć fakt, że przewidziano w nich tylko badanie niskopoziomowych cech takich jak barwa czy tekstura, a nie uwzględniono związku między różnobarwnymi obszarami, które w rzeczywistości tworzą jeden obiekt. Ponadto wiele metod preferuje obrazy nadmiernie lub niedostatecznie posegmentowane i wreszcie istnieją nieskorygowane dysproporcje zakresów wartości miar składowych (np. między miarą wewnętrznej jednorodności poszczególnych regionów i miarą różnicy między różnymi regionami).

Mimo tych wad, istniejące obecnie metody nienadzorowane najlepiej nadają się do wybierania najlepszej segmentacji spośród uzyskanych za pomocą tego samego algorytmu, ale z różnymi parametrami [127]. Właśnie tego typu nienadzorowane metody oceny segmentacji zastosowano w niniejszej pracy.

4.3 Ocena jakości segmentacji obrazów monochromatycznych

Poniżej przedstawiono wybrane miary pozwalające na ocenę jakości obrazów monochromatycznych. Charakteryzują one stopień podobieństwa między segmentacją obrazu a oryginałem. Przyjęto, że w obrazie posegmentowanym klasy są reprezentowane przez średnią wartość pikseli z obrazu oryginalnego. W obydwu przypadkach, dopuszczalne zakresy wartości pikseli są identyczne. Wszystkie prezentowane poniżej miary nadają się do oceny segmentacji zawierających więcej niż dwie klasy.

W poniższych równaniach zastosowano następujące oznaczenia: I – obraz oryginalny, I_a – segmentacja obrazu, w której klasy reprezentowane są przez średnią wartości pikseli w danej klasie, I_d – obraz różnicy, $\bar{I}_d$ – średnia wartość obrazu różnicy, D – zbiór wszystkich punktów obrazu, x – piksel, n – liczba pikseli, k – liczba poziomów szarości w obrazie oryginalnym, m – liczba klas w ostatecznej segmentacji, $p_I(i)$, $p_{I_a}(i)$ – prawdopodobieństwo wystąpienia piksela o wartości i w obrazie I czy I_a, $p_{II_a}(i,j)$ – łączne prawdopodobieństwo odcieni szarości w obrazach I i I_a, H_I, H_{I_a} – entropie obrazów I i I_a.

Pierwsza miara jakości to *zmodyfikowane odchylenie przeciętne* obrazu oryginalnego od segmentacji obliczane jako średnia z wartości bezwzględnych odchyleń punktów obrazu od średniej danej klasy. Różni się tym od typowego odchylenia przeciętnego [55], że dla każdej klasy obliczana jest osobna średnia. Zmodyfikowane odchylenie przeciętne wyraża się wzorem:

$$a_d = \frac{1}{n} \sum_{x \in D} |I(x) - I_a(x)| \,. \tag{4.1}$$

Druga miara wykorzystuje obraz różnicy obliczany według wzoru: $I_d(x) = I(x) - I_a(x)$. Jest to *odchylenie standardowe obrazu różnicy*, które w

pewien sposób charakteryzuje jednorodność klas uzyskanych w wyniku segmentacji. W przypadku, gdy każda z klas jest prawie jednorodna, σ_d przyjmuje wartości bliskie 0. Natomiast, w przypadku klas zawierających wiele różnorodnych wartości dodatnie. Odchylenie standardowe obrazu różnicy dane jest wzorem:

$$\sigma_d = \sqrt{\sum_{x \in D} \frac{\left(I_d(x) - \bar{I}_d\right)^2}{(n-1)}}. \tag{4.2}$$

Trzecią miarą jest *błąd średniokwadratowy*, który tak jak poprzednie funkcje, oparty jest na różnicy między oryginałem a obrazem zmodyfikowanym. Najszersze zastosowanie znalazł w kompresji obrazów[46] oraz w stegoanalizie. Można go również zastosować do oceny jakości segmentacji. Wyraża się on wzorem:

$$MSE = \frac{1}{n} \sum_{x \in D} \left(I(x) - I_a(x)\right)^2. \tag{4.3}$$

Ze względu na to, że większe błędy (większe odchylenia od średniej) mają większą wagę [8], miara ta pozwala na określenie, w której segmentacji istnieje wiele punktów znacząco różniących się od średnich poszczególnych klas. Tak jak poprzednie miary, błąd średniokwadratowy przyjmuje wartości nieujemne.

Informacja wzajemna jest miarą zależności pomiędzy dwiema zmiennymi losowymi. Do jej najważniejszych własności można zaliczyć, iż jest nieujemna $M(I, I_a) > 0$ oraz to, że jest symetryczna $M(I, I_a) = M(I_a, I)$. Informacja wzajemna jest pojęciem wywodzącym się z teorii informacji. Często, tak jak entropię, wyraża się ją w bitach. W literaturze podaje się równoważne definicje informacji wzajemnej, które jednak pozwalają interpretować to pojęcie na różne sposoby [82].

Pierwsza definicja dana równaniem:

$$M(I, I_a) = H(I) - H(I|I_a), \tag{4.4}$$

gdzie $H(I)$:

$$H(I) = -\sum_{i=0}^{k} p_I(i) \log p_I(i), \tag{4.5}$$

a $H(I|I_a)$:

$$H(I|I_a) = -\sum_{i=0}^{k} \sum_{j=0}^{m} p_{II_a}(i) \frac{\log p_{II_a}(i,j)}{p_{I_a}(j)} \tag{4.6}$$

najlepiej wyjaśnia znaczenie terminu informacja wzajemna [82]. W powyższym równaniu $H(I)$ oznacza entropię obrazu I a $H(I|I_a)$ entropię warunkową opartą na prawdopodobieństwie warunkowym. Gdy entropię traktuje się jako miarę niepewności, wartość obliczoną za pomocą równania 4.4 można interpretować jako informację o tym, o ile zmniejsza się niepewność dotycząca obrazu I, gdy obraz I_a jest znany lub ile informacji o obrazie I jest zawartej w obrazie I_a.

Drugie sformułowanie informacji wzajemnej dane jest równaniem [82]:

$$M(I, I_a) = H(I) + H(I_a) - H(I, I_a), \tag{4.7}$$

gdzie $H(I, I_a)$ wynosi:

$$H(I, I_a) = -\sum_{i=0}^{k}\sum_{j=0}^{m} p_{II_a}(i,j) \log p_{II_a}(i,j). \tag{4.8}$$

W odróżnieniu od pierwszej postaci, zamiast entropii warunkowej pojawia się tutaj entropia łączna $H(I_a, I)$. Równanie 4.7 ujawnia przewagę informacji wzajemnej nad entropią łączną. Jak widać informacja wzajemna zależy nie tylko od łącznego rozkładu prawdopodobieństwa, ale też od zawartości informacyjnej poszczególnych obrazów. Jest to szczególnie istotne w zastosowaniach takich jak dopasowywanie obrazów [63, 82], ponieważ zmniejsza szanse na takie ustawienie dopasowywanych obrazów, w którym pokrywające się fragmenty są bardzo podobne, ale nie zawierają żadnych informacji [82].

Ostatnia postać, jest związana z dywergencją Kullbacka-Leiblera, która służy do określana rozbieżności pomiędzy dwoma rozkładami prawdopodobieństwa. Postać ta określona jest równaniem:

$$M(I, I_a) = \sum_{i=0}^{k}\sum_{j=0}^{m} p_{II_a}(i,j) \log \frac{p_{II_a}(i,j)}{p_I(i) \cdot p_{I_a}(j)} \tag{4.9}$$

Wynika z niej, że informacja wzajemna jest miarą odległości pomiędzy łącznym rozkładem prawdopodobieństwa odcieni szarości w obrazach, a przypadkiem gdy obrazy są całkowicie niezależne tzn. gdy $p_{II_a}(i,j) = p_I(i) \cdot p_{I_a}(j)$ [82].

Mimo że informacja wzajemna znalazła najszersze zastosowanie w dopasowywaniu obrazów medycznych pochodzących z różnych urządzeń np. tomografu komputerowego i rezonansu magnetycznego, to można jej również użyć do porównywania wyników segmentacji z oryginalnym obrazem. Wartość informacji wzajemnej powinna osiągać największą wartość, gdy klasy w obrazie najlepiej odpowiadają obiektom przedstawionym w obrazie oryginalnym. W prezentowanym rozwiązaniu informacja wzajemna pozwala określić ilość informacji o obrazie zawartej w jego segmentacji. Dodatkowo zaletą informacji wzajemnej jest to, że nie musi istnieć bezpośredni związek między wartościami w porównywanych obrazach - nie muszą być porównywane przykładowo obrazy: oryginalny oraz zawierający średnie klas. W pracy [112] obliczano informację wzajemną dla obrazów, z których jeden zawierał arbitralne etykiety regionów. Umożliwiło to rozróżnienie punktów o tym samym odcieniu szarości należących do różnych obiektów.

Znormalizowanym wariantem informacji wzajemnej jest *redundancja* dana wzorem:

$$R_{I,I_a} = \frac{M_{I,I_a}}{H_I + H_{I_a}}. \tag{4.10}$$

Osiąga ona minimum wynoszące zero, gdy zawartość obrazów jest niezależna. Wartość maksymalną osiąga gdy znajomość jednego z obrazów staje się zbędna.

4.4 Dobór optymalnych parametrów analizy skupień dla obrazów monochromatycznych

W przypadku metod, w których można zmieniać parametry, jakość segmentacji z reguły zależy od doboru tych parametrów. Ponieważ proponowane podejście pozwala na zmianę metody klasteryzacji, miary podobieństwa, zestawu atrybutów oraz sposobu ich standaryzacji czy normalizacji przyprowadzono eksperyment pozwalający na stwierdzenie jaki zestaw parametrów zapewnia uzyskanie najlepszej segmentacji. W poniższym porównaniu wykorzystano cztery różnie algorytmy grupowania (opisane w punkcie A.3): metodę pojedynczego wiązania (SLINK), metodę pełnego wiązania (CLINK), metodę średniego wiązania (UPGMA) oraz metodę Warda. Każdą z nich testowano ze wszystkimi metodami standaryzacji/normalizacji opisanymi w punkcie A.5. Dodatkowo dla każdej metody przeprowadzono test bez żadnej standaryzacji. Na potrzeby próby wybrano dwanaście zestawów atrybutów, które tworzyły wektor cech zlewiska. Zestawy były następujące:

1. wartość średnia pikseli zlewiska,
2. wartość średnia oraz odchylenie standardowe pikseli w poszczególnych zlewiskach,
3. wartość średnia, odchylenie standardowe pikseli oraz rozmiar zlewiska liczony w pikselach,
4. wartość średnia oraz wariancja pikseli w poszczególnych zlewiskach,
5. wartość średnia, wariancja pikseli oraz rozmiar zlewiska liczony w pikselach,
6. minimalna wartość piksela występująca w zlewisku,
7. maksymalna wartość piksela występująca w zlewisku,
8. wartość średnia oraz rozstęp wartości pikseli zlewiska,
9. wartość średnia oraz minimalna pikseli zlewiska,
10. wartość średnia oraz maksymalna pikseli zlewiska,
11. wartości minimalna oraz maksymalna pikseli zlewiska,
12. wartość średnia, minimalna, maksymalna oraz rozstęp wartości pikseli zlewiska.

Rodzaje atrybutów opisano szczegółowo w punkcie 3.3.1. Metody SLINK, CLINK i UPGMA wymagają współczynnika, który pozwoli na określenie podobieństwa obiektów. Wykorzystano siedem następujących współczynników podobieństwa/

miar odległości: odległość euklidesową, średnią odległość euklidesową, współczynnik różnicy kształtu, współczynnik kosinusowy, współczynnik korelacji liniowej Pearsona, metrykę Canberry i współczynnik Bray'a-Curtisa. Wszystkie wymienione współczynniki opisano szczegółowo w części A.4 dodatku. Metoda Warda posiada "wbudowaną" miarę podobieństwa, tak więc nie można było jej używać ze współczynnikami wymienionymi powyżej. Wszystkie możliwe kombinacje metody grupowania, metod standaryzacji/normalizacji oraz zestawu atrybutów poddano testom na czterech obrazach medycznych (rysunek 4.3) – jednym pochodzącym z tomografu komputerowego (CT) i trzech z rezonansu magnetycznego (MRI PD, T1, T2).

Wszystkie obrazy poddano przetwarzaniu wstępnemu. Na początku każdy obraz poddano działaniu filtru rozmywającego zachowującego krawędzie (filtr symulujący proces dyfuzji, której prędkość zależy od lokalnej wartości gradientu [80], bardziej szczegółowy opis umieszczono w puncie 3.2.1). Dla wszystkich obrazów przyjęto następujące parametry rozmywania – *liczba iteracji:* 5, $\Delta t = 0,12$, $K = 0,3$. Następnie za pomocą filtru gradientowego określono położenie krawędzi. Wynik działania filtru gradientowego poddano progowaniu. W przypadku obrazu CT wartość progu wynosiła 5%, a w przypadku obrazów MRI 2% maksymalnej wartości piksela występującej na danym obrazie. Dla każdego zestawu parametrów wygenerowano segmentacje zawierające od 2 do 12 klas. W ten sposób otrzymano w sumie 58 080 segmentacji. Wszystkie zostały poddane ocenie za pomocą pięciu miar jakości segmentacji opisanych w punkcie 4.3: zmodyfikowanego odchylenia przeciętnego, błędu średniokwadratowego, odchylenia standardowego obrazu różnicy, informacji wzajemnej oraz redundancji. Następnie dla każdej segmentacji zawierającej konkretną liczbę klas znaleziono najlepszą wartość (minimum lub maksimum) poszczególnych funkcji oceny. Dalej określono, które zestawy parametrów pozwoliły uzyskać dobrą segmentację obrazu przy zadanej liczbie klas. Za dobre segmentacje uznano, te którym przynajmniej jedna z miar jakości przyznała ocenę nieróżniącą się o więcej niż 10% od najlepszego wyniku. Parametry segmentacji tzn. metoda grupowania, standaryzacja/normalizacja oraz zestaw atrybutów posortowano według liczby przypadków, w której dany zestaw dał dobry wynik segmentacji. Zestawy o tej samej liczbie dobrych segmentacji posortowano według wartości *zmodyfikowanego odchylenia przeciętnego* (równanie 4.1). Parametry, które pozwoliły na uzyskanie, maksymalnej liczby, 11 dobrych wyników zamieszczono w tabeli 4.1. Oznaczenia użyte w tabeli są skróconymi nazwami metod i współczynników przedstawionych w dodatku A. Wyjątkiem od tej reguły są oznaczenia metod standaryzacji. Ich znaczenie jest następujące: S – równanie A.21, L1 – równanie A.24, L01 – równanie A.25 a LS1 – równanie A.26 [dodatek]. Wybrane segmentacje przedstawiono na rysunkach 4.4 oraz 4.5.

Tabela 4.1: Zestawy parametrów analizy skupień, które najczęściej pozwalały uzyskać najlepsze wyniki segmentacji obrazów w skali szarości.

Lp.	**Najlepsze** a_d	**Metoda grupowania**	**Wsp. podobieństwa**	**Metoda standaryzacji**	**Zestaw atrybutów zlewiska**
obraz: CT, *progowanie*: 5% maksimum					
1	74	Ward	brak	L1	śr., max.
2	75	UPGMA	odl. euk.	brak	śr., odch. std.
3	75	UPGMA	śr. odl. euk	brak	śr., odch. std.
4	76	Ward	brak	L01	śr., max.
5	76	UPGMA	Canberra	L1	śr.
6	76	UPGMA	Canberra	LS1	śr.
7	76	UPGMA	Canberra	brak	śr.
8	76	UPGMA	wsp. B-C	L1	śr.
9	76	UPGMA	wsp. B-C	LS1	śr.
10	76	UPGMA	wsp. B-C	brak	śr.
11	77	Ward	brak	S	śr., max.
12	77	Ward	brak	L1	śr., min., max.
13	77	CLINK	wsp. B-C	brak	śr., min., max.
14	79	Ward	brak	L01	śr., min., max.
15	79	Ward	brak	L01	śr., min., max., rozst.
16	79	UPGMA	wsp. B-C	LS1	śr., min.
17	79	Ward	brak	S	śr., min., max.
18	79	UPGMA	wsp. B-C	L1	śr., min.
19	79	UPGMA	wsp. B-C	brak	śr., min.
20	79	Ward	brak	L1	śr., min., max., rozst.
21	79	UPGMA	wsp. B-C	brak	śr., odch. std.
22	81	CLINK	wsp. B-C	L1	śr., war.
23	88	CLINK	wsp. B-C	LS1	śr., min., max.
24	101	UPGMA	cos.	L1	śr., max.
25	101	UPGMA	cos.	LS1	śr., max.
26	101	UPGMA	cos.	brak	śr., max.
27	108	CLINK	cos.	brak	śr., min., max.
28	111	CLINK	cos.	LS1	śr., min., max.
obraz: MRI PD , *progowanie*: 2% maksimum					
1	22	Ward	brak	L1	śr.

ciąg dalszy na następnej stronie...

Tabela 4.1 – ciąg dalszy

Lp.	**Najlepsze** a_d	**Metoda grupowania**	**Wsp. podobieństwa**	**Metoda standaryzacji**	**Zestaw atrybutów zlewiska**
2	22	Ward	brak	L01	śr.
3	22	Ward	brak	brak	śr.
4	25	Ward	brak	L1	śr., war.
5	25	Ward	brak	L01	śr., war.
6	26	Ward	brak	brak	śr., min.
7	26	Ward	brak	LS1	śr., max.
8	26	Ward	brak	LS1	śr., min.
obraz: MRI T1, *progowanie*: 2% maksimum					
1	13	Ward	brak	brak	śr., odch. std., rozm.
2	13	CLINK	różn. kszt.	brak	śr., odch. std., rozm.
3	13	CLINK	odl. euk.	brak	śr., odch. std., rozm.
4	13	CLINK	śr. odl. euk	brak	śr., odch. std., rozm.
5	14	CLINK	odl. euk.	brak	śr., max.
6	14	CLINK	śr. odl. euk	brak	śr., max.
7	14	UPGMA	odl. euk.	brak	śr., odch. std., rozm.
8	14	UPGMA	śr. odl. euk	brak	śr., odch. std., rozm.
9	16	CLINK	różn. kszt.	S	śr., odch. std., rozm.
10	17	UPGMA	różn. kszt.	S	śr., odch. std., rozm.
11	17	Ward	brak	LS1	śr., odch. std., rozm.
12	19	UPGMA	korel.	L1	śr., odch. std., rozm.
13	19	UPGMA	cos.	L1	śr., odch. std., rozm.
14	19	CLINK	cos.	L1	śr., odch. std., rozm.
15	20	CLINK	korel.	L1	śr., odch. std., rozm.

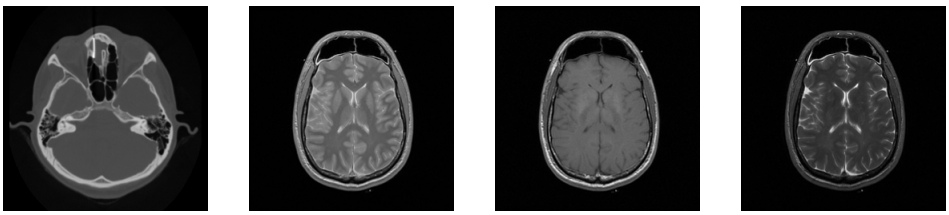
ciąg dalszy na następnej stronie...

Tabela 4.1 – ciąg dalszy

Lp.	**Najlepsze** a_d	**Metoda grupowania**	**Wsp. podobieństwa**	**Metoda standaryzacji**	**Zestaw atrybutów zlewiska**
16	21	CLINK	odl. euk.	S	śr., war., rozm.
17	21	CLINK	śr. odl. euk	S	śr., war., rozm.
obraz: MRI T2, *progowanie*: 2% maksimum					
1	16	CLINK	Canberra	L1	śr.
2	16	CLINK	Canberra	LS1	śr.
3	16	CLINK	Canberra	brak	śr.
4	16	CLINK	wsp. B-C	L1	śr.
5	16	CLINK	wsp. B-C	LS1	śr.
6	16	CLINK	wsp. B-C	brak	śr.
7	18	CLINK	wsp. B-C	L1	śr., min.
8	18	CLINK	wsp. B-C	brak	śr., min.
9	19	CLINK	odl. euk.	brak	śr., odch. std., rozm.
10	19	CLINK	śr. odl. euk	brak	śr., odch. std., rozm.
11	19	Ward	brak	brak	śr., odch. std., rozm.
12	21	CLINK	wsp. B-C	L1	śr., war.
13	22	CLINK	wsp. B-C	brak	śr., odch. std.

Wyniki przedstawione w tabeli 4.1 pozwalają stwierdzić, że optymalne parametry analizy skupień zależą od typu obrazu, w którym usuwana jest nadsegmentacja. Można jednak wysnuć kilka ogólnych wniosków. W przypadku trzech obrazów testowych (CT, PD, T1) metoda minimalnej wariancji Warda znalazła się na pierwszym miejscu listy optymalnych parametrów. Jednak tylko w przypadku obrazu PD pojawiała się na liście najczęściej. Metoda średniego wiązania (UPGMA) najczęściej daje dobre rezultaty w przypadku obrazów CT. Daje ona również dobre wyniki w przypadku skanów MRI T1, ale nie tak często jak metoda metoda pełnego wiązania (CLINK). Podczas usuwania nadsegmentacji w obrazach typu MRI T2 najlepiej sprawdza się metoda pełnego wiązania (CLINK) - tylko w jednym spośród najlepszych zestawów pojawiła się metoda Warda. Metoda pojedynczego wiązania (SLINK) pozwalała na uzyskanie dobrych segmentacji w przypadku wszystkich obrazów testowych jednak dużo rzadziej niż pozostałe trzy metody. Z tego względu nie pojawiła się ona w tabeli 4.1.

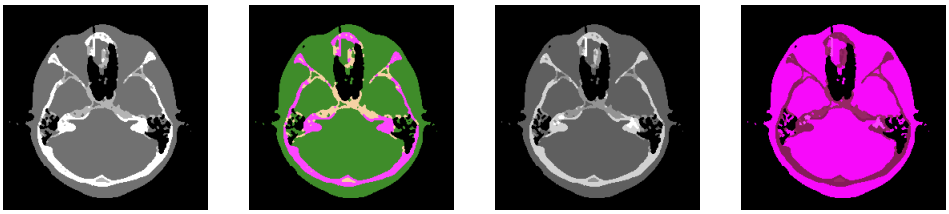
obrazy testowe CT i MRI



(a) obraz CT *bitów na piksel:* 12 | (b) obraz PD *bitów na piksel:* 11 | (c) obraz T1 *bitów na piksel:* 10 | (d) obraz T2 *bitów na piksel:* 11

Rysunek 4.3: Obrazy monochromatyczne o rozmiarze 256 × 256, dla których szukano optymalnych parametrów analizy skupień (*obraz CT*: pakiet danych przykładowych dla biblioteki VTK, *obrazy MRI*: zestaw obrazów m_vm10xx z projektu Visible Human).

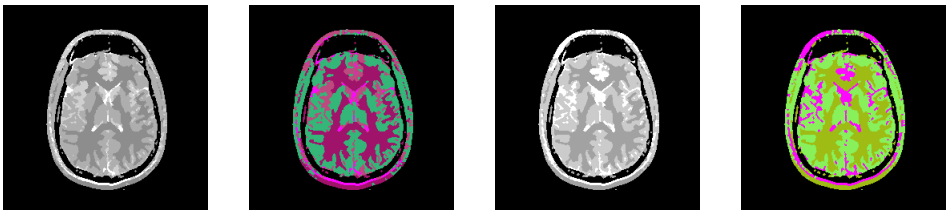
CT - headsq



(a) *metoda:* Warda, *wsp.:* wbudowany, *std.:* L1, *atrybuty:* śr., max., *klas:* 4 | (b) *metoda:* UPGMA, *wsp.:* odl. euk., *std.:* brak, *atrybuty:* śr., odch. std., *klas:* 5

Rysunek 4.4: Wybrane segmentacje obrazów z tomografu komputerowego (CT). Każdą segmentację przedstawiono w odcieniach szarości (poszczególne klasy oznaczono średnią wartością pikseli w tej klasie) oraz za pomocą pseudokoloru (parametry oryginalnego obrazu – *rozmiar*: 256 × 256, *bitów na piksel*: 12, *zestaw obrazów*: pakiet danych przykładowych dla biblioteki VTK).

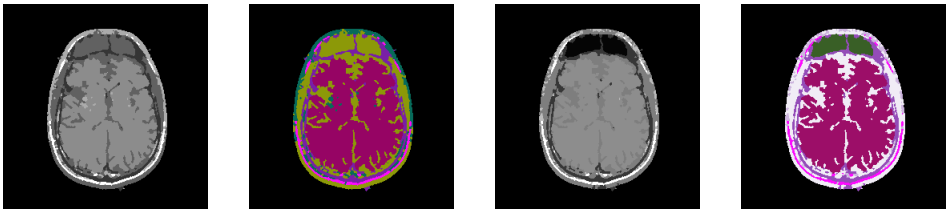
MRI PD - mri06



(c) *metoda:* Warda, *wsp.:* wbudowany, *std.:* L1, *atrybuty:* śr., *klas:* 5

(d) *metoda:* Warda, *wsp.:* wbudowany, *std.:* brak, *atrybuty:* śr., *klas:* 4

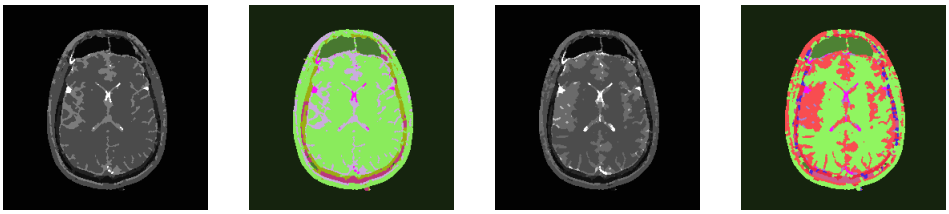
MRI T1 - mri06



(e) *metoda:* Warda, *wsp.:* wbudowany, *std.:* brak, *atrybuty:* śr., odch. std., rozm., *klas:* 6

(f) *metoda:* CLINK, *wsp.:* różn. kszt., *std.:* brak, *atrybuty:* śr., odch. std., rozm., *klas:* 6

MRI T2 - mri06



(g) *metoda:* CLINK, *wsp.:* Canberra, *std.:* L1, *atrybuty:* śr., *klas:* 8

(h) *metoda:* CLINK, *wsp.:* Bray-Curtis, *std.:* L1, *atrybuty:* śr., min., *klas:* 11

Rysunek 4.5: Wybrane segmentacje obrazów z rezonansu magnetycznego (MRI). Każdą segmentację przedstawiono w odcieniach szarości (poszczególne klasy oznaczono średnią wartością pikseli w tej klasie) oraz za pomocą pseudokoloru (parametry obrazów oryginalnych – *rozmiar*: 256×256, *bitów na piksel*: 11 (PD i T2), 10 (T1)), *zestaw obrazów*: m_vm10xx z projektu Visible Human).

W większości dobrych segmentacji, metody CLINK oraz UPGMA – czyli metody wymagające zewnętrznego współczynnika podobieństwa – były używane w połączeniu ze współczynnikiem Bray'a-Curtisa, a w dalszej kolejności ze współczynnikiem kosinusowym, (średnią) odległością euklidesową, współczynnikiem Canberry, współczynnikiem różnicy kształtu oraz współczynnikiem korelacji. Jak wspomniano wcześniej metoda Warda nie używa "zewnętrznej" miary podobieństwa. Rodzaj najlepszego współczynnika (przy pominięciu metody Warda) również zależy od typu segmentowanego obrazu. Dla obrazu CT najlepszym współczynnikiem okazała się odległość euklidesowa, dla obrazu MRI T1 – współczynnik różnicy kształtu, a dla MRI T2 - metryka Canberry.

Optymalne zestawy atrybutów, podobnie jak metody i współczynniki podobieństwa, różnią się w zależności od segmentowanego obrazu. Można jednak zauważyć kilka prawidłowości:

1. W przypadku obrazów CT oraz MRI typu T1 najlepsze zestawy zawierają więcej niż jeden atrybut. Oznacza to, że zastosowanie większej liczby atrybutów może poprawić jakość segmentacji.

2. Rozstęp wartości nie ma dużego znaczenia przy usuwaniu nadsegmentacji w obrazach medycznych CT oraz MRI, ponieważ wystąpił tylko w dwóch spośród najlepszych zestawów atrybutów. Rozmiar zlewiska liczony w pikselach okazał się również przydatny tylko w przypadku obrazów T1 i T2.

3. Większość najlepszych zestawów atrybutów zawiera więcej niż jeden atrybut zlewiska. Najczęściej średnia sprawdzała się w połączeniu z odchyleniem standardowym.

Najlepsze segmentacje uzyskiwano za pomocą następujących zestawów atrybutów:

1. średnia wartość pikseli w zlewisku oraz ich odchylenie standardowe,

2. średnia wartość pikseli zlewiska,

3. średnia, minimalna oraz maksymalna wartość pikseli zlewiska.

4. średnia oraz maksymalna wartość pikseli zlewiska,

5. średnia oraz minimalna wartości pikseli zlewiska,

6. średnia wartość oraz wariancja pikseli zlewiska,

7. średnia wartość oraz wariancja pikseli zlewiska i rozmiar zlewiska.

Standaryzacja atrybutów nie okazała się konieczna do uzyskania bardzo dobrych wyników, ale może pomóc w uzyskaniu dobrej segmentacji, gdy atrybuty różnią się rzędem wielkości (np. średnia oraz wariancja). Najczęściej, wśród

dobrych parametrów analizy skupień pojawiała się jedna z metod określonych równaniami A.24 (L1), A.25 (L01) oraz A.26 (LS1) [dodatek].

4.5 Ocena jakości segmentacji obrazów barwnych

Do oceny jakości segmentacji obrazów barwnych nie można bezpośrednio zastosować wzorów wymienionych w punkcie 4.3. Wynika to z faktu, że piksele w takich obrazach opisane są za pomocą wektorów (zazwyczaj 3-elementowych) wartości. Poniżej przedstawiono, modyfikacje, których należało dokonać, aby przystosować poprzednio wykorzystane funkcje oceny do obrazów barwnych. Składowe barwne oznaczono indeksem c, a zbiór zawierający wszystkie składowe oznaczono S. Na przykład I_d^c jest składową c obrazu różnicy. Zmieniono również znaczenie symbolu k we wzorze na informację wzajemną. Oznacza on teraz liczbę różnych barw występujących w obrazie.

Pierwszą z przystosowanych metod oceny jest *zmodyfikowane odchylenie przeciętne*. Wyrażone wzorem:

$$a_d = \sum_{x \in D} \sum_{c \in S} \left| I_d^c(x) \right| / (n \cdot c), \tag{4.11}$$

gdzie: $I_d^c(x) = I^c(x) - I_a^c(x)$. W przypadku obrazów barwnych sumowane są odchylenia składowych od odpowiadających im obrazów średnich we wszystkich punktach obrazu. Przy obliczaniu średniego odchylenia uwzględniono, oprócz liczby punktów, także liczbę składowych.

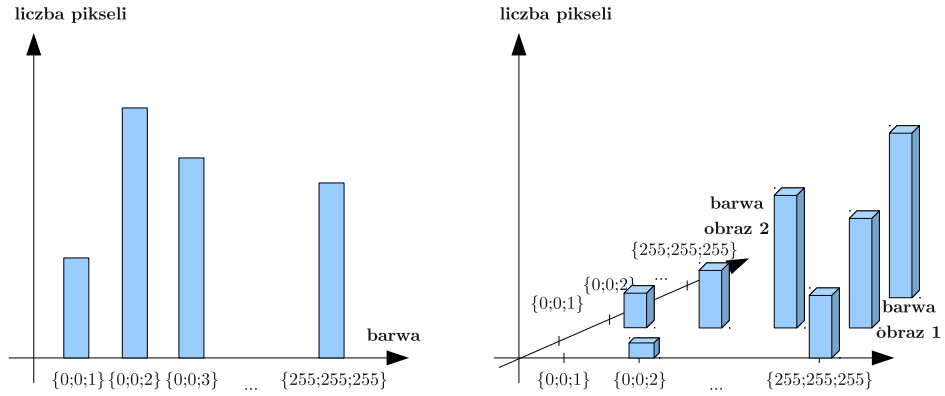
Modyfikacje drugiej z zastosowanych miar, czyli *błędu średniokwadratowego*, są podobne. Sumy kwadratów odchyleń we wszystkich punktach obliczane są dla wszystkich składowych i odpowiednich obrazów ze średnimi klas. Przy obliczaniu średniej również uwzględniono liczbę składowych.

$$MSE = \sum_{x \in D} \sum_{c \in S} \left(I^c(x) - I_a^c(x) \right)^2 / (n \cdot c). \tag{4.12}$$

Trzecią funkcją, zastosowaną do oceny segmentacji obrazów monochromatycznych było *odchylenie standardowe obrazu różnicy*. Zastosowane modyfikacje były takie same jak w przypadku dwóch poprzednich funkcji.

$$\sigma_d = \sqrt{\sum_{x \in D} \sum_{c \in S} \frac{\left(I_d^c(x) - \bar{I}_d^c \right)^2}{(n \cdot c)}}. \tag{4.13}$$

Dwie kolejne funkcje czyli *informacja wzajemna* i jej znormalizowana postać – *redundancja* nie zostały zmodyfikowane. Jednak aby obliczyć ich wartości potrzebne są histogramy oryginalnego obrazu I oraz obrazu średnich I_a. Histogramy te wykorzystywane są z kolei do wyznaczenia prawdopodobieństw występujących we wzorach 4.9 i 4.10. W przypadku obrazów barwnych zazwyczaj oblicza



Rysunek 4.6: Przykładowe histogramy obrazów barwnych wyznaczane podczas obliczania informacji wzajemnej i redundancji.

się osobny histogram dla każdej ze składowych. Takie podejście nie jest odpowiednie do obliczania informacji wzajemnej oraz redundancji. Dlatego podczas obliczania tych miar dla obrazów barwnych, każdą kombinację wartości składowych traktowano jako całość, jako jeden symbol. Innymi słowy, w przypadku obrazów RGB jednowymiarowe histogramy posiadały słupki dla następujących barw: $\{0;0;1\}, \{0;0;2\}, \ldots, \{255;255;255\}$. Histogramy łączne posiadały słupki dla każdej możliwej pary barw występujących w obrazie oryginalnym i w segmentacji. Sposób wyznaczania histogramów zilustrowano na rysunku 4.6. W rzeczywistej implementacji funkcji oceny pamięć rezerwowano tylko dla barw (par barw), które faktycznie występowały w przetwarzanych obrazach. Pozwoliło to na znaczne zmniejszenie wymagań pamięciowych.

Oprócz metod opisanych powyżej istnieją takie, które powstały z myślą o ocenie segmentacji obrazów barwnych. Poniżej opisano trzy metody tego typu. W pracy [62] Liu i Yang zaproponowali funkcję F, która pozwala na ilościową ocenę jakości segmentacji, a jednocześnie nie wymaga podawania żadnych parametrów wejściowych. Według autorów uwzględnia ona kryteria dobrej segmentacji sformułowane przez Shapiro i Haralicka [41] bezpośrednio lub pośrednio, z wyjątkiem czwartego postulatu dotyczącego krawędzi. Funkcja F, realizuje kryterium trzecie bezpośrednio, przyznając gorsze oceny segmentacjom, które zawierają wiele małych regionów – co może oznaczać według autorów – że regiony zawierają wiele małych dziur. Kryterium pierwsze i drugie jest realizowane pośrednio poprzez uwzględnienie w funkcji F błędu barwy e_i (określonego równaniem 4.16). Duży błąd barwy e_i oznacza, według Liu i Yanga, dużą wariancję wartości pikseli wewnątrz regionu, a więc brak jednorodności i co za tym idzie, niezgodność z pierwszym postulatem. Sposób, w który uwzględniono drugie kryterium dobrej segmentacji jest najmniej oczywisty. Jeżeli sąsiadujące ze sobą regiony są do siebie podobne, to ich suma będzie miała mały błąd barwy e_i, a więc algorytm

segmentacji powinien połączyć je w jeden region. Jeżeli tego nie zrobi, wynik jego działania zostanie gorzej oceniony, ponieważ segmentacja będzie zawierała więcej mniejszych regionów (funkcja F preferuje większe regiony).

Funkcja F została ustalona empirycznie i ma postać [62]:

$$F(\mathbf{I}) = \sqrt{R} \cdot \sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}}, \tag{4.14}$$

gdzie: R – liczba klas, na które został podzielony obraz, A_i – pole i-tej klasy, e_i – średni błąd barwy w i-tej klasie.

Powyższa funkcja przyjmuje wartości nieujemne $F(\mathbf{I}) \geqslant 0$, przy czym mniejsze wartości odpowiadają lepszym segmentacjom, a 0 oznacza segmentację idealną. Na wartość funkcji oceny F mają wpływ dwa składniki. Pierwszy składnik $\sqrt{R}$ jest globalnym współczynnikiem kary, który jak wspomniano, zapewnia przyznanie gorszej oceny segmentacjom ze zbyt dużą liczbą klas R, a drugi składnik, czyli $\sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}}$, ma charakter lokalny, ponieważ odnosi się do poszczególnych regionów. Regiony, które posiadają duży błąd barwy e_i, podobnie jak regiony o niewielkim polu powierzchni powodują zwiększenie wartości F, a więc pogorszenie oceny.

Dodatkowo autorzy w [62] wprowadzili czynnik $\frac{1}{1000 \cdot N \cdot M}$, który zmniejsza zakres wartości oceny oraz normalizuje jej wartość przez rozmiar obrazu (N i M to szerokość i wysokość obrazu). Ostatecznie zastosowana funkcja F miała postać:

$$F(\mathbf{I}) = \frac{1}{1000 \cdot N \cdot M} \cdot \sqrt{R} \cdot \sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}}. \tag{4.15}$$

Wykorzystany w równaniu błąd barwy e_i dany jest wzorem [62, 127]:

$$e_i = \sum_{x \in R_i} \sqrt{\sum_{s \in S} (I_s(x) - I_{sa}(x))^2}, \tag{4.16}$$

gdzie: S – zbiór wszystkich składowych obrazu, R_i – i-ta klasa, I_s – składowa s obrazu oryginalnego, I_{sa} – składowa s segmentacji, w której klasy reprezentowane są przez średnie wartości pikseli.

M. Borsotti, P. Campadelli oraz R. Schettini [23] przeprowadzili eksperyment z zastosowaniem funkcji F. Najlepsza segmentacja została prawidłowo rozpoznana przez funkcję oceniającą, jednak drugie miejsce otrzymała segmentacja określona przez autorów jako "czysty szum". Składała się z wielu małych regionów. Doprowadziło to do spostrzeżenia, że gdy liczba regionów jest bardzo duża, to czynnik $\sqrt{R}$ przestaje mieć znaczenie ponieważ jest mnożony przez sumę błędów barwy e_i w poszczególnych regionach, a te są równe bądź bliskie zeru. Zaproponowali zmodyfikowaną funkcję F' następującej postaci [23, 76]:

$$F'(\mathbf{I}) = \frac{1}{10000(N \cdot M)} \cdot \sqrt{\sum_{A=1}^{Max} [R(A)]^{1+1/A}} \cdot \sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}}. \tag{4.17}$$

We wzorze 4.17 zmieniła się postać współczynnika kary: zamiast $\sqrt{R}$ pojawił się $\sqrt{\sum_{A=1}^{Max}[R(A)]^{1+1/A}}$ gdzie: $R(A)$ jest liczbą regionów o rozmiarze równym A. Wykładnik $1+\frac{1}{A}$ dla $A=1$ wynosi 2, dla $A=2$ wynosi $1,5$ a dla $A=10000$ wynosi $1,0001$, tak więc gdy segmentacja składa się głównie z dużych regionów, to wyrażenie pod pierwiastkiem jest zbliżone do R. Gdy jednak segmentacja zawiera wiele małych regionów wykładnik $1+\frac{1}{A}$ powoduje zwiększenie wartości fuknkji oceny F' – a więc jej pogorszenie. Niestety podobnie jak F, także funkcja F' osiąga wartość minimalną (zero) dla obrazów, które nie są posegmentowane tzn. piksele nie są połączone w klasy i każdy z nich stanowi osobną klasę - mimo że liczba klas R jest duża, to błąd barwy dla każdej z nich wynosi zero i ostateczna wartość funkcji również wynosi zero. Problem ten wynika, z faktu, że obie funkcje składają się z dwóch podstawowych czynników.

Drugim problemem zauważonym przez Borsottiego i in. jest to, że funkcja podobnym segmentacjom przyznaje zbyt różniące się oceny – przyczyną jest to, że nowy czynnik znajduje się poza sumą mierzącą błędy barwy. Dlatego autorzy zaproponowali [23, 76] jeszcze jedną funkcję oceny jakości segmentacji daną równaniem:

$$Q(\mathbf{I}) = \frac{1}{10000(N \cdot M)} \cdot \sqrt{R} \cdot \sum_{i=1}^{R} \left[\frac{e_i^2}{1+logA_i} + \left(\frac{R(A_i)}{A_i} \right)^2 \right]. \tag{4.18}$$

Funkcja Q tak jak F i F' składa się z trzech głównych czynników. Pierwszy z nich, podobnie jak w przypadku poprzednich funkcji, zapewnia zmniejszenie zakresu wartości funkcji oceny i jej normalizację względem rozmiaru analizowanego obrazu. Drugi, czyli $\sqrt{R}$ jest współczynnikiem kary zwiększającym wartość oceny, gdy segmentacja zawiera wiele regionów. Ostatni czynnik jest bardziej rozbudowany niż w przypadku pozostałych funkcji. Położono w nim większy nacisk na karanie regionów o dużym błędzie barwy, oraz regionów o niewielkiej powierzchni. Za pierwszą część kary odpowiada składnik $\frac{e_i^2}{1+logA_i}$ – bo przyjmuje duże wartości dla regionów niejednorodnych, ponieważ charakteryzują się dużym błędem barwy e_i. Dodatkowo zastąpienie $\sqrt{A_i}$ przez $1+logA_i$ powoduje silniejsze karanie takich regionów. Za drugą część odpowiada składnik $\left(\frac{R(A_i)}{A_i}\right)^2$ powoduje pogorszenie oceny dla segmentacji ze zbyt dużą liczbą regionów. Jeżeli cały obraz o rozmiarach 256×256 pikseli zostanie podzielony na regiony o rozmiarze 4 pikseli, wtedy wartość tego składnika wynosić będzie 4096^2. W przypadku, gdy regiony będą większe i będą składały się z 256 pikseli, wtedy składnik będzie równy tylko 1. Oznacza to, że przyjmuje on duże wartości, gdy w segmentacji znajduje się wiele małych regionów o tym samym rozmiarze. W ten sposób funkcja Q dodatkowo "karze" nadsegmentację. Wprowadzenie tego składnika spowodowało również to, że w odróżnieniu od F i F', Q nie osiąga wartości minimalnej dla obrazu nieposegmentowanego (czyli obrazu, w którym każdy piksel stanowi osobną

klasę). Według Borsotti'ego i in. oceny przyznawane przez funkcję Q najlepiej odpowiadają wzrokowej ocenie jakości segmentacji [23].

4.6 Dobór optymalnych parametrów analizy skupień dla obrazów barwnych

W przypadku obrazów barwnych. tak jak i dla monochromatycznych wykonano porównanie różnych parametrów analizy skupień w celu sprawdzenia, czy istnieją takie, które pozwalają na uzyskanie najlepszych wyników w usuwaniu nadsegmentacji. Próbne segmentacje były wykonywane na pięciu różnych obrazach testowych przedstawionych na rysunkach: 4.7a, 4.7d, 4.7g, 4.8a oraz 4.8d. Porównanie jest podobne do tego, które wykonano dla obrazów monochromatycznych. Tutaj również wykorzystano cztery różne algorytmy grupowania (opisane szczegółowo w punkcie A.3): metodę pojedynczego wiązania, metodę pełnego wiązania, metodę średniego wiązania oraz metodę Warda. Każdą metodę testowano ze wszystkimi metodami standaryzacji/normalizacji wymienionymi w punkcie A.5, a dodatkowo przeprowadzono test bez żadnej standaryzacji.

Należy zaznaczyć, że atrybuty obliczane dla składowych RGB podczas standaryzacji atrybutów były grupowane. W szczególności: średnia atrybutu ($\bar{x}_i$), jego odchylenie standardowe (σ_i), wartość minimalna ($x_{min\,i}$), wartość maksymalna ($x_{max\,i}$), a także suma wartości ($\sum_c \sum_{i=1}^{t} x_{ij}$) – występujące w równaniach A.21 - A.26 – były obliczane wspólnie dla wszystkich składowych. Przykładowo w przypadku atrybutu *wartość średnia* obliczonego dla obrazu RGB, $x_{min\,i}$ będzie minimalną wartością średnią pikseli w zlewisku występującą we wszystkich składowych. Efektem takiego postępowania jest jednakowa zmiana skali atrybutów obliczonych dla poszczególnych składowych. Tak więc wszystkie średnie skalowane są razem, wartości odchylenia standardowego skalowane są razem, itd. Grupowanie atrybutów wprowadzono, aby standaryzacja nie zniekształcała informacji o barwie. Nie zmienia to faktu, że standaryzacja w dalszym ciągu niweluje różnice w zakresie wartości między atrybutami np. między średnią a wariancją. Atrybuty były grupowane wyłącznie podczas standaryzacji. W pozostałych etapach analizy skupień poszczególne atrybuty obliczone dla poszczególnych składowych były traktowane osobno.

Zestawy atrybutów, które charakteryzowały każde zlewisko, były takie same jak w przypadku obrazów monochromatycznych. Każdy rodzaj atrybutu był obliczany dla wszystkich składowych obrazu RGB. Dlatego wektory atrybutów zlewiska zawierały trzy razy więcej składowych niż w przypadku obrazów monochromatycznych. Zestawy atrybutów opisano szczegółowo w punkcie 4.4.

Do określania podobieństwa obiektów wykorzystano siedem współczynników podobieństwa/miar odległości. Wszystkie współczynniki były takie same jak w przypadku obrazów monochromatycznych (punkt 4.4). Opisano je szczegółowo w

punkcie A.4. Wspomniane miary podobieństwa były niezbędne przy korzystaniu z metod UPGMA, CLINK oraz SLINK. Nie były one potrzebne tylko w metodzie Warda, która posiada "wbudowaną" miarę podobieństwa.

Przed segmentacją wszystkie obrazy były poddane przetwarzaniu wstępnemu. Każdy obraz został poddany działaniu selektywnego filtru rozmywającego opartego na metodzie poziomic, w której prędkość przemieszczania linii o stałej jasności zależy od zmodyfikowanej miary krzywizny tej linii [123]. Przeprowadzona, modyfikacja skutkowała zmniejszeniem prędkości poziomicy w miejscach, w których moduł gradientu obrazu ma dużą wartość. Ponieważ stopień rozmycia jest zależny od prędkości linii, małe obszary (duża krzywizna krawędzi) nie różniące się znacząco od swoich sąsiadów (niewielki gradient) są rozmywane szybko. Najwolniej natomiast rozmywane są krawędzie dużych obszarów o dużym kontraście. Dla wszystkich obrazów przyjęto następujące parametry rozmywania – *liczba iteracji:* 5, $\Delta t = 0,12$, $K = 0,3$. Bardziej szczegółowy opis filtru znajduje się punkcie 3.2.1. Następnie zastosowano jeden z filtrów gradientowych przedstawionych w punkcie 2.5.2. Moduł gradientu poddano progowaniu, a następnie działaniu transformacji wododziałowej. Płytkie zlewiska, tzn. takie których głębokość nie przekraczała pewnego progu nazywanego *poziomem łączenia*, są scalane z sąsiadami (punkt 3.2.3). Zastosowanie filtru rozmywającego, progowanie jak również łączenie płytkich zlewisk miało na celu wstępne zmniejszenie nadsegmentacji. Parametry przetwarzania wstępnego dobrano, tak aby sprowadzić liczbę zlewisk do wartości zbliżonej do 2000 – 3000, ale równocześnie, tak aby nie zostały połączone zlewiska należące do różnych obiektów. Wartości tych parametrów podano w opisach obrazów na rysunkach 4.7 oraz 4.8. Próg oraz poziom łączenia podano jako procent maksymalnej wysokości w obrazie gradientowym. Określono również rodzaj filtru gradientowego, jak również liczbę zlewisk w nadmiernie posegmentowanym obrazie.

Dla każdego zestawu parametrów i dla każdego obrazu wygenerowano segmentacje zawierające od 2 do 20 klas. Daje to w sumie 125 400 segmentacji. Wszystkie zostały ocenione za pomocą ośmiu miar jakości segmentacji opisanych w punkcie 4.5: zmodyfikowanego odchylenia przeciętnego, błędu średniokwadratowego, odchylenia standardowego obrazu różnicy, informacji wzajemnej, redundancji, funkcji F, F' oraz Q. Następnie, podobnie jak w przypadku obrazów monochromatycznych, znaleziono dla określonego obrazu i określonej liczby klas najlepszą ocenę (w zależności od funkcji – wartość minimalną lub maksymalną). Gdy najlepsze oceny dla danej liczby klas i danego obrazu były znane – wybrano dobre segmentacje. Przyjęto, że dobra segmentacja to taka, której przynajmniej jedna miara jakości przyznała ocenę nie różniącą się o więcej niż 10% od najlepszego wyniku. Dalej obliczono ile razy dane parametry tzn. metoda klasteryzacji, równanie standaryzujące, zestaw atrybutów oraz współczynnik podobieństwa/odległość pozwoliły uzyskać dobrą segmentację bez względu na liczbę

klas w wynikowej segmentacji. Zestawy parametrów, które pozwoliły uzyskać 19 (czyli maksymalną liczbę) dobrych segmentacji zamieszczono w tabeli 4.2. Zostały one posortowane według najlepszej wartości funkcji oceny Q uzyskanej dla danego zestawu (innymi słowy przyjęto wartość Q dla optymalnej liczby klas wybranej z zakresu $2 \div 20$). Oznaczenia zastosowane w tabeli 4.2 są identyczne jak w przypadku tabeli 4.1. Przykładowe segmentacje poszczególnych obrazów testowych przedstawiono na rysunkach 4.7 oraz 4.8 – spośród zestawów w tabeli 4.2 wybrano ten, który otrzymał minimalną (najlepszą) ocenę od funkcji Q. Optymalną liczbę klas w prezentowanych segmentacjach również wybrano kierując się funkcją Q.

Tabela 4.2: Zestawy parametrów analizy skupień, które najczęściej pozwalały uzyskać najlepsze wyniki segmentacji obrazów barwnych.

Lp.	**Najlepsze Q**	**Metoda grupowania**	**Wsp. podobieństwa**	**Metoda standaryzacji**	**Zestaw atrybutów zlewiska**
			obraz: kwiatek_3		
1	2279	Ward	brak	brak	śr., odch. std.
2	2441	Ward	brak	brak	śr., war.
3	3518	CLINK	cos.	S	śr.
4	3980	CLINK	odl. euk.	brak	śr., odch. std.
5	3980	CLINK	śr. odl. euk	brak	śr., odch. std.
6	6436	CLINK	Canberra	S	min.
7	6540	CLINK	wsp. B-C	S	min.
8	7266	CLINK	wsp. B-C	S	śr., min., max., rozst.
			obraz: mur_1		
1	309	Ward	brak	brak	śr., war.
2	317	Ward	brak	L1	śr.
3	357	CLINK	wsp. B-C	brak	śr., war.
4	384	CLINK	wsp. B-C	L1	śr.
5	384	CLINK	wsp. B-C	LS1	śr.
6	384	CLINK	wsp. B-C	brak	śr.
7	414	CLINK	odl. euk.	LS1	śr.
8	414	CLINK	śr. odl. euk	LS1	śr.
9	427	CLINK	wsp. B-C	brak	śr., odch. std.
			obraz: Lena		

ciąg dalszy na następnej stronie...

Tabela 4.2 – ciąg dalszy

Lp.	**Najlepsze Q**	**Metoda grupowania**	**Wsp. podobieństwa**	**Metoda standaryzacji**	**Zestaw atrybutów zlewiska**
1	204	Ward	brak	brak	śr., war.
2	243	Ward	brak	L1	śr.
3	252	CLINK	odl. euk.	L1	śr., min., max.
4	252	CLINK	śr. odl. euk	L1	śr., min., max.
5	262	CLINK	odl. euk.	L01	śr., min., max.
6	262	CLINK	śr. odl. euk	L01	śr., min., max.
7	266	CLINK	odl. euk.	S	śr., min., max.
8	266	CLINK	śr. odl. euk	S	śr., min., max.
9	275	CLINK	odl. euk.	brak	śr., min., max.
10	275	CLINK	śr. odl. euk	brak	śr., min., max.
11	279	CLINK	rożn. kszt.	S	śr.
12	279	CLINK	rożn. kszt.	brak	śr.
13	281	CLINK	rożn. kszt.	brak	śr., war.
14	319	Ward	brak	L1	śr., min.
15	319	Ward	brak	brak	śr., min.
16	333	Ward	brak	L01	śr., min., max.
17	359	CLINK	odl. euk.	L01	śr., min., max., rozst.
18	359	CLINK	śr. odl. euk	L01	śr., min., max., rozst.
19	694	CLINK	wsp. B-C	L1	min.
			obraz: samolot		
1	56	Ward	brak	brak	śr., war.
2	62	Ward	brak	brak	śr.
3	64	Ward	brak	S	śr.
4	68	Ward	brak	L1	śr.
5	70	Ward	brak	L01	śr.
6	86	Ward	brak	S	śr., max.
7	90	Ward	brak	brak	śr., odch. std.
			obraz: słonie		
1	482	Ward	brak	L1	śr.
2	517	Ward	brak	S	śr.
3	522	Ward	brak	LS1	śr.
4	580	Ward	brak	L01	śr., min.
5	586	Ward	brak	brak	śr., war.
6	601	Ward	brak	L01	śr., war.

ciąg dalszy na następnej stronie...

Tabela 4.2 – ciąg dalszy

Lp.	Najlepsze Q	Metoda grupowania	Wsp. podobieństwa	Metoda standaryzacji	Zestaw atrybutów zlewiska
7	602	Ward	brak	L01	śr.
8	644	CLINK	wsp. B-C	LS1	śr., min.

Wyniki przedstawione w tabeli 4.2 pozwalają na wysnucie pewnych ogólnych wniosków dotyczących parametrów segmentacji, umożliwiających uzyskanie dobrej segmentacji zgodnie z kryteriami wbudowanymi w funkcje oceny).

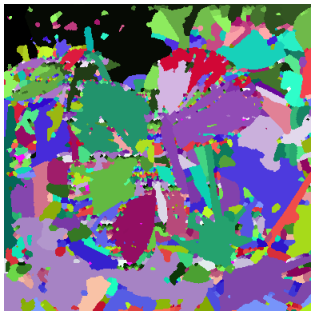
Segmentacje przedstawione na obrazach 4.7c, 4.7f, 4.7i, 4.8c oraz 4.8f otrzymały następujące wartości funkcji Q: 2280 w przypadku obrazu *kwiatek_3*, 309 dla obrazu *mur_1*, 205 dla obrazu *Lena*, 57 otrzymała segmentacja obrazu *samolot* i 482 w wypadku obrazu *słonie*. Najgorsze (czyli największe) wartości funkcji Q obliczone dla segmentacji obrazów testowych wyniosły odpowiednio: 93528, 29649, 24010, 4739 i 23968. Wartości te obliczono dla segmentacji powstałych przy wykorzystaniu innych niż optymalne parametry analizy skupień, jednak zawierających tę samą liczbę klas, co segmentacje przedstawione na obrazach: 4.7c, 4.7f, 4.7i, 4.8c oraz 4.8f. Porównanie odpowiednich wartości wyraźnie pokazuje, że wyniki segmentacji silnie zależą od doboru parametrów analizy skupień.

Jak wynika z tabeli 4.2 w przypadku obrazów *mur_1*, *Lena* i *samolot*, zestaw parametrów analizy skupień pozwalający na uzyskanie najlepszej segmentacji to: (1) metoda minimalnej wariancji Warda, (2) brak standaryzacji i (3) zestaw atrybutów składający się ze średniej zlewiska oraz jego wariancji. Warto również zauważyć, że w przypadku obrazu *kwiatek_3* zestaw ten pojawia się na drugim miejscu. Segmentacja uzyskana za jego pomocą otrzymała od funkcji Q ocenę wynoszącą 2441 podczas gdy segmentacja uzyskana przy pomocy najlepszego zestawu otrzymała ocenę 2280. Jest to niewielka różnica biorąc pod uwagę, że najgorsza wartość funkcji Q dla obrazu *kwiatek_3* wynosi 93528. Sytuacja jest podobna w przypadku obrazu *słonie*. Zestaw parametrów składający się z metody minimalnej wariancji Warda, średniej oraz wariancji zlewiska jest piąty wśród najlepszych zestawów przedstawionych w tabeli 4.2. Trzeba zauważyć, że różnica między wartościami funkcji Q wynosi, w przybliżeniu, 100. Jest to niewielka różnica biorąc pod uwagę, że najgorsza, uzyskana dla tego obrazu, wartość funkcji Q to 23968. Uzyskane wyniki oznaczają, że dla obrazu testowego *słonie* istnieje kilka zestawów atrybutów pozwalających na uzyskanie segmentacji o podobnej, dobrej jakości.

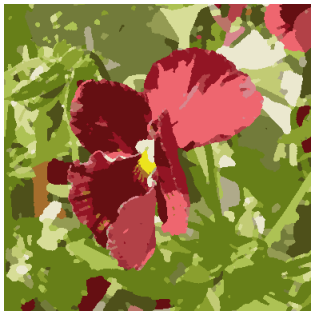
Trudno oczekiwać, że identyczne parametry będą dawały najlepsze wyniki dla wszystkich obrazów. Jednak na podstawie wyników z tabeli 4.2 można powiedzieć, że w przypadku obrazów barwnych metoda minimalnej wariancji Warda, w

kwiatek_3

(a)
obraz oryginalny,
rozmiar: 384 × 384
pikseli



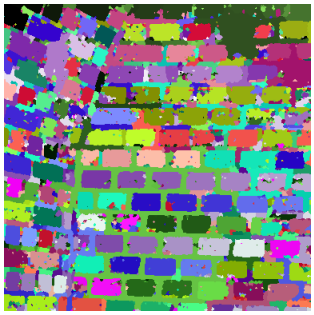
(b)
gradient: skp, *próg:* 5%,
poziom łączenia: 10%
zlewisk: 1965



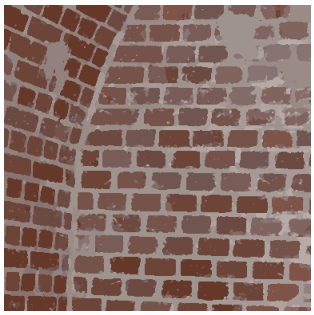
(c)
met.: Warda,
wsp: brak, *std.:* brak,
atr.: śr., odch. std.
klas: 19

mur_1

(d)
obraz oryginalny,
rozmiar: 384 × 384
pikseli



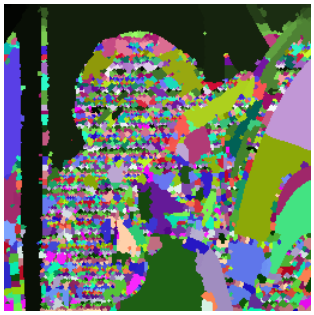
(e)
gradient: pca, *próg:* 0%,
poziom łączenia: 8%
zlewisk: 2555



(f)
met.: Warda,
wsp: brak, *std.:* brak,
atr.: śr., war. *klas:* 18

Lena

(g)
obraz oryginalny,
rozmiar: 256 × 256
pikseli



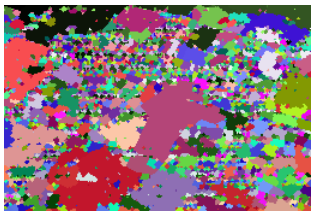
(h)
gradient: skp, *próg:* 1%,
poziom łączenia: 10%
zlewisk: 3164

(i)
met.: Warda, *std.:* brak,
atr.: śr., war. *klas:* 20

Rysunek 4.7: Wybrane segmentacje obrazów barwnych. W pierwszej kolumnie przedstawiono obrazy oryginalne (źródło obrazu *Lena*: `http://www.ii.uni.wroc.pl/~anl/dyd/PO/examples1/`), w drugiej obrazy nadmiernie posegmentowane (po wstępnej redukcji nadsemgentacji) natomiast w trzeciej wynik redukcji nadsegmentacji za pomocą analizy skupień.

samolot

(a)
obraz oryginalny,
rozmiar: 256 × 171
pikseli (obraz: 37073
z [67])

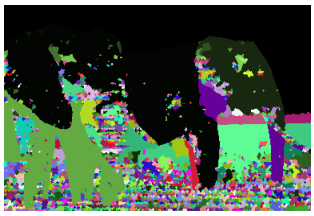


(b)
gradient: pca, *próg:* 1%,
poziom łączenia: 1%
zlewisk: 2985

(c)
met.: Warda, *std.:* brak,
atr.: śr., war. *klas:* 20

słonie

(d)
obraz oryginalny,
rozmiar: 384 × 256
pikseli (obraz: 296059
z [67])



(e)
gradient: pca, *próg:* 0%,
poziom łączenia: 6%
zlewisk: 3107

(f)
met.: Warda, *std.:* L1,
atr.: śr. *klas:* 11

Rysunek 4.8: Wybrane segmentacje obrazów barwnych. W pierwszej kolumnie przedstawiono obrazy oryginalne, w drugiej obrazy nadmiernie posegmentowane (po wstępnej redukcji nadsemgentacji) natomiast w trzeciej wynik redukcji nadsegmentacji za pomocą analizy skupień. (źródło obrazów *samolot* oraz *słonie* – *The Berkeley Segmentation Dataset* [67]).

połączeniu z brakiem standaryzacji oraz z zestawem atrybutów składających się ze średniej i wariancji zlewiska pozwala na uzyskanie segmentacji dobrej jakości z niewielką liczbą klas.

Warto również zauważyć, że metoda pełnego wiązania (CLINK) w połączeniu z odległością euklidesową lub współczynnikiem Bray'a-Curtisa i zestawem atrybutów zawierającym więcej niż jeden atrybut daje wyniki, które zazwyczaj nie różnią się znacznie od najlepszych. Taka sytuacja miała miejsce w przypadku czterech obrazów testowych (wszystkich z wyjątkiem obrazu *samolot*). Oprócz wspomnianej odległości euklidesowej innymi, rzadziej pojawiającymi się w połączeniu z metodą CLINK, współczynnikami były: metryka Canberry, współczynnik kosinusowy, średnia odległość euklidesowa oraz współczynnik różnicy kształtu.

Fakt, iż metoda pojedynczego (SLINK) oraz średniego wiązania (UPGMA)

nie pojawiają się w tabeli 4.2, nie oznacza, że te metody są bezużyteczne. Jak pokazano w punkcie 4.4 metoda UPGMA pozwala na uzyskanie dobrych segmentacji monochromatycznych obrazów CT lub MRI T1. Ponadto, jak zostanie pokazane w rozdziale 6, metoda SLINK może zostać użyta przy tworzeniu segmentacji według innych kryteriów niż te uwzględnione przez zastosowane funkcje oceny.

Oprócz zestawu atrybutów składającego się ze średniej oraz wariancji zlewiska, dobre segmentacje można uzyskać poprzez zastosowanie podobnego zestawu składającego się ze średniej oraz odchylenia standardowego. Można zastosować również wyłącznie średnią zlewiska. Warto jednak podkreślić, że dla czterech spośród pięciu obrazów testowych najlepsze wyniki uzyskano dzięki zestawom składającym się z więcej niż jednego atrybutu. W tabeli 4.2 pojawiają się również zestawy zawierające wartość minimalną, maksymalną, rozmiar zlewiska, a także, dość rzadko, rozstęp wartości zlewiska.

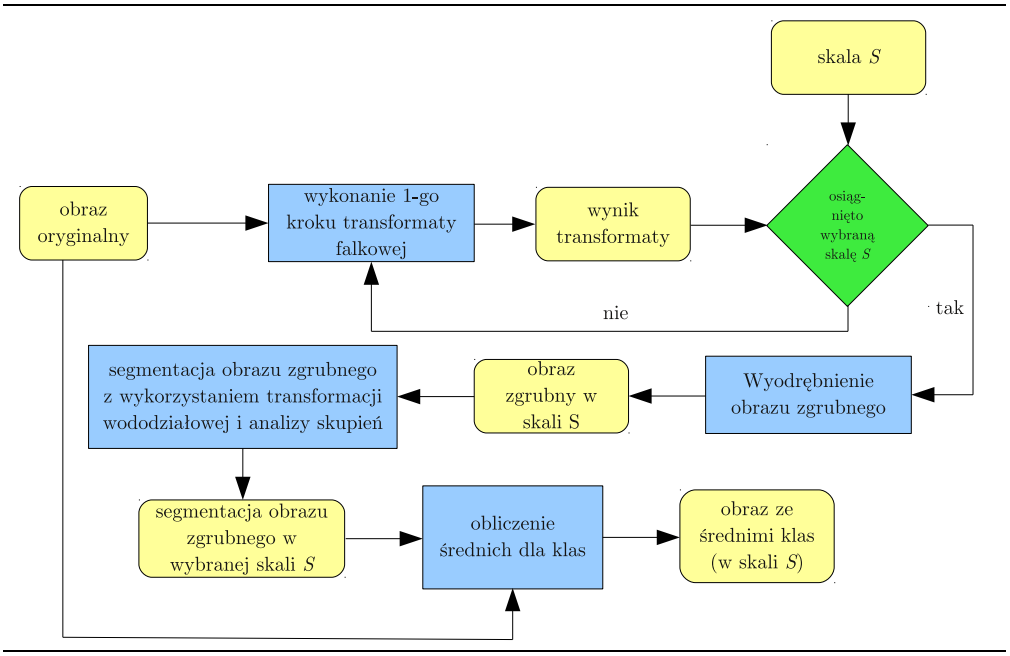
Ostatnią kwestią jest wpływ standaryzacji na otrzymane wyniki. Jak już wspomniano nie występuje ona w żadnym zestawie parametrów, który daje dobre wyniki w przypadku wszystkich obrazów. Porównując zestawy atrybutów, które różnią się tylko rodzajem standaryzacji (obraz *mur_1* zestawy 4–6, obraz *Lena* zestawy 3–10, obraz *samolot* zestawy 2–5, obraz *słonie* zestawy 1–3 oraz 5–6) można stwierdzić, że wpływ standaryzacji zastosowanych atrybutów na jakość segmentacji jest zazwyczaj niewielki. Wynika to z faktu, że standaryzowane wartości nie różnią się znacząco rzędem wielkości.

Rozdział 5

Poprawa wydajności metody

W przypadku proponowanej metody czas analizy skupień w dużym stopniu zależy od liczby badanych obiektów, czyli zlewisk w nadmiernie posegmentowanym obrazie. Liczba ta zależy, z kolei, od przyjętego sposobu przetwarzania wstępnego obrazu (filtru rozmywającego, filtru gradientowego, parametrów wspomnianych filtrów) oraz od treści obrazu. W rozdziale 2 przedstawiono podstawowe sposoby redukcji nadsegmentacji. Pokazano, że *gradient-pca* jest mniej wrażliwy na szum oraz niewielkie szczegóły przedstawione na obrazie niż *gradient-skp* (punkt 2.5.2), przez co prowadzi do mniejszej nadsegmentacji. Progowanie obrazu gradientowego przed poddaniem go transformacji wododziałowej pozwala usunąć niewielkie wartości gradientu oraz zredukować liczbę zlewisk w ostatecznej segmentacji (punkt 3.2.2). Również zalewanie płytkich zlewisk kontrolowane przez *poziom łączenia* pozwala na osiągnięcie podobnego efektu (punkt 3.2.3). Niestety, nawet po zastosowaniu powyższych sposobów, w przypadku rzeczywistego obrazu o rozmiarach 256×256 pikseli liczba zlewisk może przekroczyć 10000. Ich łączenie za pomocą analizy skupień jest długotrwałe, a ponadto powoduje problemy z obliczaniem, przechowywaniem i przetwarzaniem macierzy podobieństwa. Wynika stąd potrzeba zmniejszenia liczby zlewisk. Jednak używając sposobów wymienionych powyżej, nie zawsze można zmniejszyć liczbę zlewisk do poziomu, w którym czas analizy skupień nie jest zbyt długi, a równocześnie wynik segmentacji zawiera wszystkie istotne przedstawione na obrazie obiekty. Zbyt wysoki próg czy poziom łączenia może spowodować połączenie zlewisk należących do dwóch różnych obiektów.

Z przeprowadzonych eksperymentów wynika, że aby całość procesu segmentacji, na współczesnym komputerze, nie trwała dużo dłużej niż 20 sekund, liczba zlewisk nie powinna przekraczać 2000. Poniżej opisano metodę pozwalającą na istotne zmniejszenie liczby zlewisk, gdy nie można inaczej uzyskać zadowalających rezultatów. Przedstawiono porównanie uzyskanych czasów segmentacji oraz



Rysunek 5.1: Diagram zmodyfikowanego algorytmu pozwalającego na szybsze uzyskanie wyników. Modyfikacja polega na zastosowaniu transformaty falkowej.

przykładowe wyniki segmentacji.

5.1 Zastosowanie transformaty falkowej

Proponowane przez autora pracy rozwiązanie [108] jest zmodyfikowaną i zaadaptowaną wersją rozwiązania przedstawionego w [48]. Jego diagram przedstawiono na rysunku 5.1. Wykorzystuje ono transformatę falkową do stworzenia piramidy obrazów o malejącej rozdzielczości. Dzięki filtrowaniu dolnoprzepustowemu związanemu z tym procesem, liczba minimów lokalnych, a co za tym idzie, liczba zlewisk zostaje zmniejszona. Na początku wykonuje się kilka kroków transformaty falkowej, a następnie dla najmniejszego obrazu zgrubnego wykonywany jest opisany wcześniej (rozdział 3) algorytm segmentacji (obliczanie obrazu gradientowego, transformacja wododziałowa, klasteryzacja oraz podział na pożądaną liczbę klas). Po przeprowadzeniu segmentacji wyniki przenoszone są za pomocą odwrotnej transformaty falkowej do oryginalnej rozdzielczości obrazu. W celu poprawy jakości segmentacji, po każdym kroku, przeprowadzana jest korekcja krawędzi, podczas której punkty leżące na brzegach obszarów przypisywane są do najbardziej podobnej klasy. W przedstawionych badaniach wykorzystano transformatę Haara.

5.2 Transformata Haara

Jednowymiarowa transformata Haara dzieli sygnał na dwie równe części. Pierwsza część składa się ze średnich dwóch kolejnych punktów. Druga część zawiera wartości szczegółów, czyli zachowuje informacje, jak oryginalne wartości punktów różnią się od ich średnich. Współczynniki dla dolnoprzepustowego $h[n]$ i górnoprzepustowego $g[n]$ filtru dla falki Harra wynoszą [48]:

$$h[n] = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right] \tag{5.1}$$

oraz

$$g[n] = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right]. \tag{5.2}$$

W przetwarzaniu obrazów wykorzystuje się dwuwymiarową transformatę Haara. Jest ona realizowana za pomocą algorytmu Mallata [18]. Pierwszy krok transformaty 2D polega na przeprowadzeniu jednego kroku transformaty 1D osobno dla każdego wiersza, a następnie dla każdej kolumny obrazu. Obraz jest zamieniany w cztery podobrazy: jeden zgrubny i trzy szczegółowe. Wszystkie obrazy są o połowę niższe i węższe niż obraz oryginalny. Obraz zgrubny C^1 jest rozmytą i pomniejszoną wersją obrazu oryginalnego, obrazy szczegółowe: D_V^1, D_H^1 i D_D^1 zawierają odpowiednio: informacje o szczegółach pionowych, poziomych i diagonalnych. Następny krok transformaty jest przeprowadzany wyłącznie na obrazie zgrubnym C^1. W wyniku tego kroku również otrzymywane są cztery obrazy: C^2, D_V^2, D_H^2 i D_D^2. Kolejne etapy transformaty będą nazywane skala 0, 1, 2, ... gdzie „skala 0" odnosi się do oryginalnego obrazu, „skala 1" do obrazu po pierwszym etapie transformaty falkowej, itd.

5.3 Zmniejszanie rozdzielczości obrazu za pomocą transformaty falkowej

Jak już wspomniano, podstawowe metody przetwarzania wstępnego obrazu: progowanie i łączenie zlewisk na podstawie ich głębokości [45], zazwyczaj nie pozwalają zredukować liczby działów wodnych bez pogorszenia jakości segmentacji (łączenia osobnych obiektów). Właśnie w tym miejscu można wykorzystać transformatę falkową do stworzenia piramidy obrazów o coraz mniejszej wielkości. Na każdym poziomie s tworzone są cztery podobrazy: C^s, D_V^s, D_H^s i D_D^s, które zastępują obraz C^{s-1}. Rozmiary nowych obrazów, to rozmiary oryginału podzielone przez 2^s). Jest to popularne rozwiązanie stosowane w przetwarzaniu obrazów [48].

Na rysunku 5.2 przedstawiono dwa pierwsze kroki transformaty falkowej. Na podstawie obrazu oryginalnego o rozdzielczości 512×512 punktów powstały obrazy C^1 o rozdzielczości 256×256 oraz C^2 o rozdzielczości 128×128 punktów.

Ponieważ obrazy C^1, C^2 powstają dzięki zastosowaniu funkcji skalującej, która ma właściwości filtru dolnoprzepustowego [48] liczba minimów lokalnych w

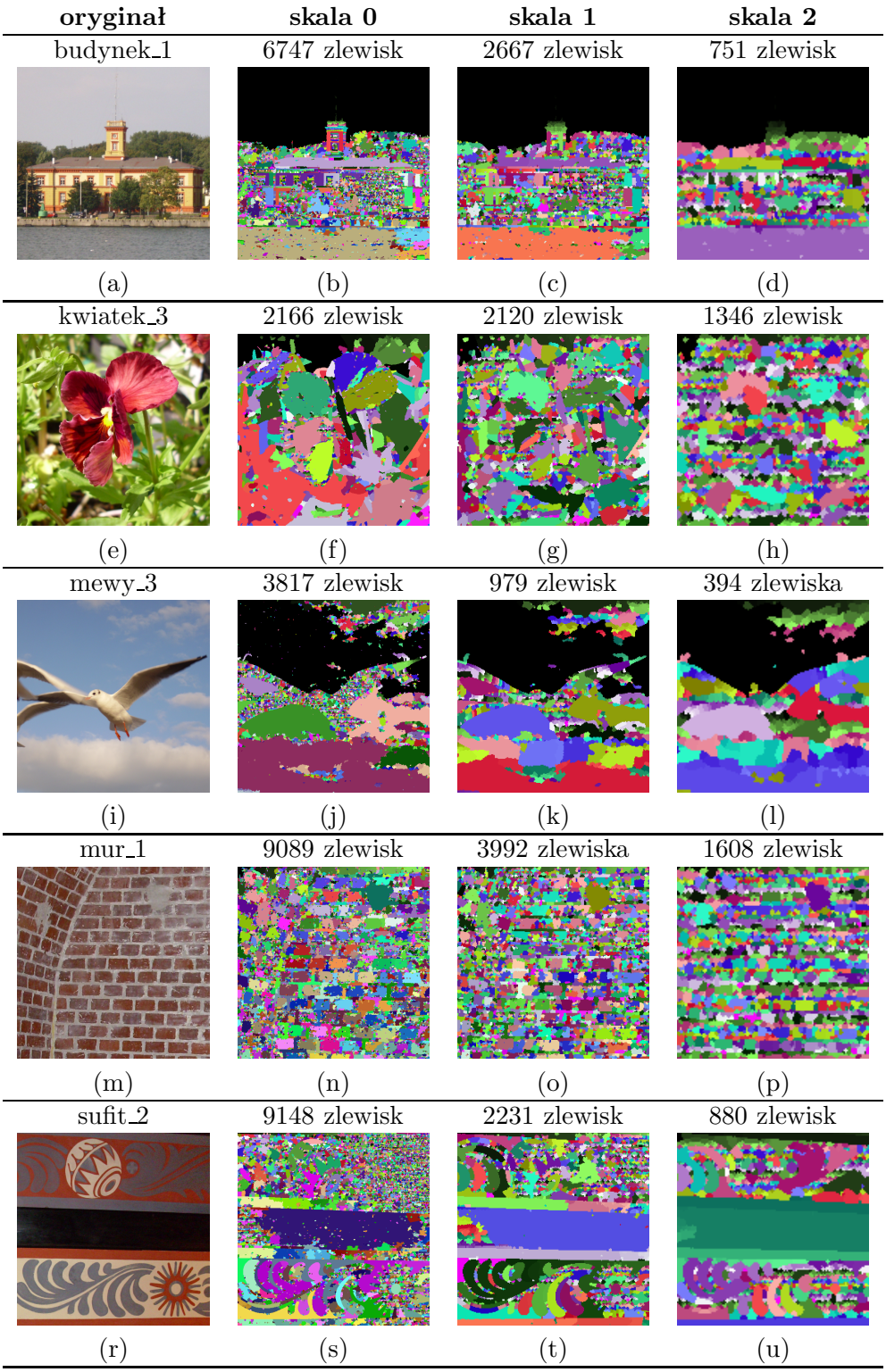
(a) (b) (c)

Rysunek 5.2: Kolejne kroki transformaty falkowej: (a) obraz oryginalny, (b) obrazy: C^1, D_V^1, D_H^1 oraz D_D^1, (c) obraz C^1 zastąpiony przez C^2, D_V^2, D_H^2, D_D^2.

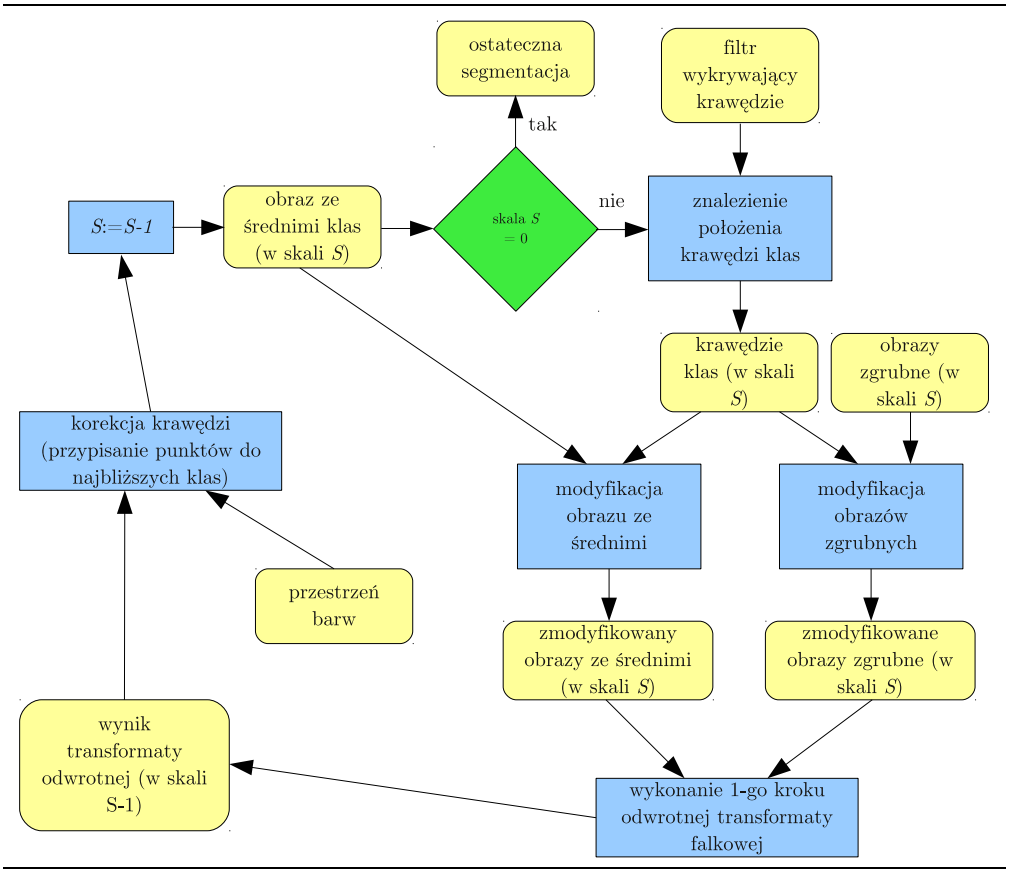
obrazach gradientowych $|\nabla C^1|$ oraz $|\nabla C^2|$ zostaje zmniejszona. Oznacza to, że również liczba zlewisk powstałych w wyniku transformacji wododziałowej będzie mniejsza. Rysunek 5.3 przedstawia wpływ skali obrazu poddawanego transformacie wododziałowej na stopień nadsegmentacji. Każdy z obrazów we wszystkich skalach poddano takiemu samemu przetwarzaniu wstępnemu (sposób przygotowania obrazów omówiono w punkcie 5.6, a parametry podano razem z wynikami segmentacji na rysunku 5.6). Jak widać, mimo dość dużego zróżnicowania obrazów testowych, wykonanie dwóch kroków transformaty falkowej pozwala na około dziewięciokrotne zmniejszenie liczby zlewisk. Wyjątkiem jest tu obraz testowy *kwiatek_3* (rysunki 5.3 e-f), w wypadku którego nastąpiła niespełna dwukrotna redukcja stopnia nadsegmentacji. Wynika to z faktu, iż zdjęcie to charakteryzuje się małą głębią ostrości i tylko obiekt pierwszoplanowy jest ostry. Pozostałe obiekty są rozmyte, co ma istotny wpływ na wynik działania filtru gradientowego i pośrednio na wynik transformacji wododziałowej.

5.4 Przenoszenie wyników segmentacji do oryginalnej rozdzielczości

Przywracanie oryginalnej rozdzielczości jest bardziej kłopotliwe niż jej zmniejszanie. Przedstawione rozwiązanie jest zmodyfikowaną wersją algorytmu opisanego w [48]. Niech obraz S^s będzie wynikiem segmentacji obrazu o niskiej rozdzielczości C^s. Składa się on z klas, których piksele mają wartości równe średniej wartości odpowiadających im punktów obrazu oryginalnego. Wartości dla każdej składowej (R,G i B) są uśredniane osobno, co oznacza, że obliczane są trzy średnie dla każdej klasy. W odróżnieniu od [48] wykorzystany w tej pracy algorytm transformacji wododziałowej nie tworzy linii działów wodnych. Ponadto, ponieważ przeniesienie wyników segmentacji do wyższej rozdzielczości odbywa się po analizie skupień, istnieje potrzeba zastosowania filtru wykrywającego krawędzie



Rysunek 5.3: Wpływ liczby wykonanych kroków transformaty falkowej na stopień nadsegmentacji obrazu.



Rysunek 5.4: Przenoszenie wyników segmentacji obrazu zgrubnego do rozdzielczości obrazu oryginalnego.

połączonych regionów (w [48] zlewiska są łączone za pomocą prostego kryterium różnicy kontrastu w obrazie o pełnej rozdzielczości). W prezentowanej pracy zastosowano dwa różne filtry wykrywające krawędzie. Zostaną one opisane bardziej szczegółowo w dalszej części rozdziału.

Pierwszym krokiem przenoszenia wyników segmentacji do pełnej rozdzielczości jest oznaczenie punktów leżących na lub blisko krawędzi regionów w obrazie S^s. Następnie tworzony jest uaktualniony obraz $S^{s'}$, w którym piksele krawędziowe zastępowane są odpowiadającymi im pikselami ze zgrubnego obrazu C^s. Formalnie:

$$S^{s'}(x,y) = \begin{cases} C^s(x,y) & \text{jeżeli } (x,y) \text{ jest punktem krawędziowym} \\ S^s(x,y) & \text{w innym wypadku.} \end{cases} \tag{5.3}$$

Następnie obrazy szczegółowe D_V^s, D_H^s, D_D^s są modyfikowane. Piksele, które nie odpowiadają punktom krawędziowym, są zerowane, podczas gdy punkty na kra-

wędziach pozostają niezmienione. Formalnie:

$$D^{s'}(x,y) = \begin{cases} D^s(x,y) & \text{jeżeli } (x,y) \text{ jest punktem krawędziowym} \\ 0 & \text{w przeciwnym wypadku.} \end{cases} \tag{5.4}$$

Gdy wszystkie obrazy w danej skali s zostaną zmodyfikowane, wykonywany jest jeden krok odwrotnej transformaty falkowej. W ten sposób powstaje obraz $S^{s-1''}$. Jeżeli piksel (x,y) był punktem krawędziowym w obrazie S^s o wartości wziętej z obrazu C^s, wtedy cztery piksele: $(2x,2y)$, $(2x+1,2y)$, $(2x,2y+1)$ i $(2x+1,2y+1)$ w obrazie $S^{s-1''}$ nie są przypisane do żadnej klasy i mają wartości równe odpowiadającym pikselom z obrazu C^{s-1}. Algorytm przegląda obraz w poszukiwaniu nieprzyporządkowanych punktów. Gdy znajdzie taki punkt, przegląda jego 8-sąsiedztwo. Za pomocą normy euklidesowej oblicza odległości tego punktu od jego sąsiadów należących już do jakiejś klasy. Piksel zostaje ostatecznie przyporządkowany do najbliższej klasy (jego wartością staje się wartość punktów w tej klasie). Zazwyczaj po pierwszym przebiegu algorytmu nie wszystkie nieprzyporządkowane punkty zostają przypisane do klas. W takim wypadku algorytm wykonuje kolejne przebiegi do momentu, w którym wszystkie punkty należą już do jakiejś klasy. Konieczność wielokrotnego przeglądania obrazu wynika z występowania krawędzi grubszych niż jeden piksel oraz pogrubiania krawędzi przez odwrotną transformatę falkową. Wynikiem algorytmu przypisywania pikseli jest obraz $S^s - 1$. Następnie wykrywane są nowe krawędzie regionów, obrazy zawierające segmentację i szczegóły są uaktualniane, wykonywany jest kolejny krok odwrotnej transformaty falkowej oraz korekcja krawędzi. Proces ten jest powtarzany do momentu osiągnięcia rozdzielczości obrazu oryginalnego.

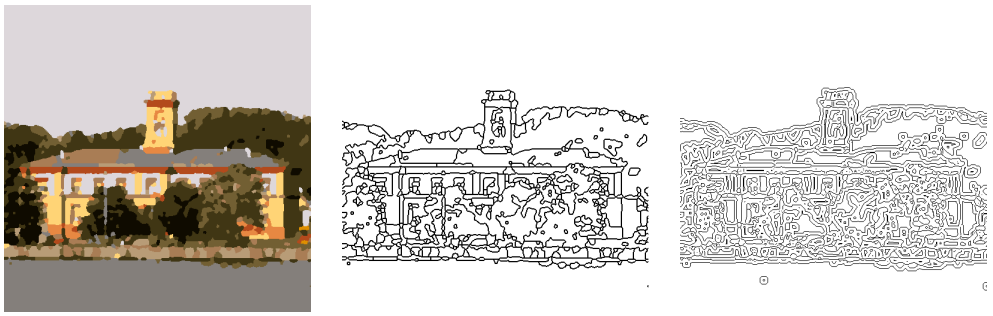
Jak wspomniano powyżej przetestowano dwie różne metody wykrywania krawędzi. Pierwszą z nich jest prosty filtr 4-sąsiedztwa. Filtr przegląda obraz z segmentacją i oznacza piksel, jeżeli jeden z jego sąsiadów ma inną wartość.

$$E^s(x,y) = \begin{cases} 1 & \text{jeżeli } \exists (x',y') \in N_4(x,y) : S^s(x',y') \neq S^s(x,y) \\ 0 & \text{w przeciwnym wypadku} \end{cases} \tag{5.5}$$

Wynikiem działania tego filtru są krawędzie o szerokości dwóch punktów, co powoduje wydłużenie czasu korekcji krawędzi. Wynik działania filtru przedstawiono na rysunku 5.5b.

Drugim przetestowanym filtrem jest filtr Canny'ego [36]. Jego działanie jest bardziej złożone niż poprzedniego algorytmu:

1. obraz rozmywany jest filtrem Gaussa,
2. tworzone są dwa nowe obrazy. Jeden z nich zawiera moduł a drugi kierunek gradientu, we wszystkich punktach obrazu oryginalnego,
3. z obrazu zawierającego moduł gradientu usuwane są wartości, które nie są maksimami lokalnym,



(a) (b) (c)

Rysunek 5.5: Porównanie filtrów krawędziowych: (a) oryginał - segmentacja obrazu *budynek_1*, (b) wynik działania filtru 4-sąsiedztwa, (c) wynik działania filtru Canny'ego.

4. obraz jest progowany z wykorzystaniem wysokiego i niskiego progu,
5. dzięki wynikom progowania można połączyć fragmenty krawędzi.

Filtr Canny'ego opisany jest bardziej szczegółowo w [36]. Do zalet filtru należy to iż tworzone przez niego krawędzie mają (zazwyczaj) szerokość jednego piksela. Niestety w przypadku sztucznych obrazów, takich jak obraz zawierający średnie klas, oprócz prawidłowych krawędzi filtr tworzy dodatkowe linie w niewielkiej odległości. Efekt ten przedstawiono na rysunku 5.5c.

5.5 Pełny algorytm segmentacji

Pełny algorytm segmentacji, uzupełniony o zmniejszanie obrazu oraz przeniesienie wyników do oryginalnej rozdzielczości, jest następujący:

1. wykonuje się s kroków transformaty falkowej (przy wykorzystaniu falki Haara), w wyniku czego powstaje następująca seria obrazów: C^1, D_V^1, D_H^1, D_D^1, ..., C^s, D_V^s, D_H^s, D_D^s,
2. obraz C^s jest poddawany działaniu selektywnego filtru rozmywającego [80],
3. obliczany jest obraz gradientowy rozmytego obrazu C^s (za pomocą *gradientu-skp* lub *gradietu-pca*),
4. wynik jest progowany,
5. przeprowadzana jest transformacja wododziałowa,
6. siadujące zlewiska są łączone na podstawie ich głębokości,
7. obliczane są wartości atrybutów zlewisk,

8. obliczana jest macierz podobieństwa/różnicy za pomocą współczynnika podobieństwa/różnicy,
9. algorytm łączy dwa najbardziej podobne klastry i uaktualnia hierarchię podobieństwa reprezentowaną przez drzewo,
10. uaktualniana jest macierz podobieństwa/różnicy,
11. jeżeli został więcej niż jeden klaster, algorytm wraca do punktu 9,
12. na podstawie hierarchii podobieństwa (drzewa) tworzona jest segmentacja S^s (w niskiej rozdzielczości) o zadanej liczbie klas, wartości pikseli w każdej z klas zastępowane są średnią wartością tej klasy,
13. wyniki segmentacji przenoszone są do oryginalnej rozdzielczości obrazu, jak opisano w punkcie 5.4.

5.6 Wyniki

Zaproponowane rozwiązanie przetestowano na pięciu różnych obrazach barwnych wszystkich o rozdzielczości 512×512 pikseli. Uzyskano je za pomocą aparatu fotograficznego pracującego w przestrzeni sRGB i w tej przestrzeni je przetwarzano. Obrazy wybrano, tak aby przedstawiały różne rodzaje obiektów. Obraz *budynek_1* (rysunek 5.3a) stanowi dość typowe zdjęcie. Obraz *kwiatek_3* (5.3e) jest zdjęciem, w którym tylko obiekt pierwszoplanowy jest ostry, a pozostałe obiekty mają niewyraźne krawędzie. Kolejny obraz *mewy_3*, oprócz lecącej mewy, zawiera również chmury czyli obiekty, które zlewają się z tłem (rysunek 5.3i). Rysunek 5.3m przedstawia zdjęcie muru - jest to obraz, w którym liczba klas, w końcowej segmentacji jest łatwa do określenia. Ostatnie zdjęcie przedstawia malowaną dekorację sufitu (obraz *sufit_2* – rysunek 5.3r) – w tym wypadku kilka dość jednolitych obszarów. Jednak na tym zdjęciu, w lewym górnym rogu, widoczne są szumy matrycy aparatu co potęguje nadsegmentację w tym obszarze.

Każdy obraz był segmentowany w skali 0, 1 i 2, co oznacza, że na wejście algorytmu segmentacji wododziałowej podawane były obrazy o rozmiarach 512×512, 256×256 i 128×128 pikseli. Pomniejszone obrazy uzyskano, wykonując jeden lub dwa kroki transformaty falkowej. Bez względu na skalę obrazy poddano przetwarzaniu wstępnemu. Zastosowano selektywny filtr rozmywający [80] (parametry: $konduktancja = 0,3$, $\Delta t = 0,12$, 5 iteracji), obliczono obraz gradientowy (*gradient-pca* lub *gradient-skp*), który następnie poddano progowaniu (próg określony jest jako wartość procentowa maksimum obrazu gradientowego). Wybór parametrów przetwarzania wstępnego oparto na wynikach przedstawionych w [104, 106]. Przygotowane obrazy poddano transformacji wododziałowej. Aby zmniejszyć liczbę zlewisk, były one łączone na podstawie ich głębokości

(punkt 3.2.3). Dział wodny jest łączony z sąsiadem, jeżeli jego głębokość nie przekracza zadanego progu – poziomu łączenia (jest on określony jako wartość procentowa największej głębokości zlewiska). Następnie przeprowadzana była analiza skupień, aby ostatecznie połączyć zlewiska w nadmiernie posegmentowanym obrazie. Przedstawione wyniki uzyskano stosując metody CLINK i UPGMA w połączeniu ze średnią i wariancją lub samą średnią jako atrybutami zlewisk. Wyniki segmentacji przenoszono do oryginalnej rozdzielczości 512×512 punktów. We wszystkich przypadkach zastosowano oba filtry wykrywające krawędzie, a odległości między punktami obliczano w przestrzeniach sRGB i CIELab. Oznacza to, że dla obrazu w danej skali istnieją cztery kombinacje filtru wykrywającego krawędzie i przestrzeni barw. W sumie uzyskano 45 segmentacji (20 segmentacji w skali 1, 20 segmentacji w skali 2 oraz 5 segmentacji bez użycia transformaty falkowej). Rysunek 5.6 przedstawia wybrane segmentacje wraz z zastosowanymi parametrami przetwarzania wstępnego, analizy skupień oraz przywracania pełnej rozdzielczości. W trakcie obliczania segmentacji mierzono czasy trwania analizy skupień oraz czas przenoszenia wyników do oryginalnej rozdzielczości. Czasy segmentacji w skali 0 przedstawiono w celach porównawczych (tabela 5.1). Jak widać użycie transformaty falkowej pozwala na znaczne zmniejszenie czasu analizy skupień zazwyczaj znacznie większy niż stopień zmniejszenia liczby zlewisk. Tylko w wypadku jednego obrazu (*kwiatek_3*) poprawa była niewielka. Wynika to z faktu, iż większość obrazu stanowi nieostre tło - powoduje to niewielką nadsegmentację i zmniejsza wpływ filtrowania dolnoprzepustowego.

Tabela 5.1: Porównanie całkowitych czasów segmentacji w zależności od liczby wykonanych kroków transformaty falkowej.

Przestrzeń barw	**Filtr krawędziowy**	**Czas analizy skupień** [s]	**Czas przywracania** [s]	**Czas całkowity** [s]	**Przyspieszenie**
obraz: budynek_1, *skala*: 0, *liczba zlewisk*: 6747					
-	-	648	-	648	-
obraz: budynek_1, *skala*: 1, *liczba zlewisk*: 2667					
sRGB	Canny	44	2	46	14,1
sRGB	4-s	44	2	46	14,1
CIELab	Canny	44	3	47	13,8
CIELab	4-s	44	4	48	13,5
obraz: budynek_1, *skala*: 2, *liczba zlewisk*: 751					
sRGB	Canny	1	3	4	162
sRGB	4-s	1	2	3	216

ciąg dalszy na następnej stronie...

Tabela 5.1 – ciąg dalszy

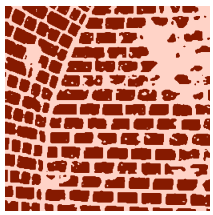
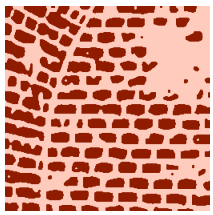
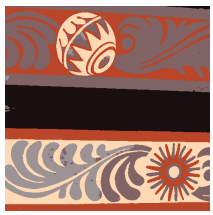
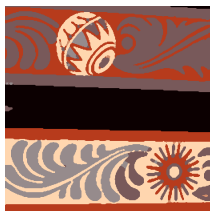
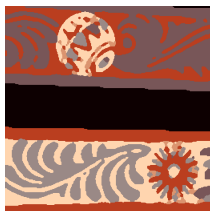
Przestrzeń barw	**Filtr krawędziowy**	**Czas analizy skupień [s]**	**Czas przywracania [s]**	**Czas całkowity [s]**	**Przyspieszenie**
CIELab	Canny	1	3	4	162
CIELab	4-s	1	5	6	108
obraz: kwiatek_3, *skala*: 0, *liczba zlewisk*: 2166					
-	-	24	-	24	-
obraz: kwiatek_3, *skala*: 1, *liczba zlewisk*: 2120					
sRGB	Canny	22	3	25	1
sRGB	4-s	22	2	24	1
CIELab	Canny	22	3	25	1
CIELab	4-s	22	4	26	0,9
obraz: kwiatek_3, *skala*: 2, *liczba zlewisk*: 1346					
sRGB	Canny	6	3	9	2,7
sRGB	4-s	6	2	8	3
CIELab	Canny	6	4	10	2,4
CIELab	4-s	6	4	10	2,4
obraz: mewy_3, *skala*: 0, *liczba zlewisk*: 3817					
-	-	138	-	138	-
obraz: mewy_3, *skala*: 1, *liczba zlewisk*: 979					
sRGB	Canny	4	2	6	23
sRGB	4-s	4	2	6	23
CIELab	Canny	4	3	7	20
CIELab	4-s	4	4	8	17
obraz: mewy_3, *skala*: 2, *liczba zlewisk*: 394					
sRGB	Canny	< 1	2	2	69
sRGB	4-s	< 1	2	2	69
CIELab	Canny	< 1	3	3	46
CIELab	4-s	< 1	4	4	35
obraz: mur_1, *skala*: 0, *liczba zlewisk*: 9089					
-	-	1500	-	1500	-
obraz: mur_1, *skala*: 1, *liczba zlewisk*: 3992					
sRGB	Canny	134	2	136	11
sRGB	4-s	134	1	135	11,1
CIELab	Canny	134	2	136	11
CIELab	4-s	134	2	136	11
obraz: mur_1, *skala*: 2, *liczba zlewisk*: 1608					
sRGB	Canny	10	2	12	125

ciąg dalszy na następnej stronie...

Tabela 5.1 – ciąg dalszy

Przestrzeń barw	**Filtr krawędziowy**	**Czas analizy skupień [s]**	**Czas przywracania [s]**	**Czas całkowity [s]**	**Przyspieszenie**
sRGB	4-s	10	2	12	125
CIELab	Canny	10	3	13	115,4
CIELab	4-s	10	4	14	107,1
obraz: sufit_2, *skala*: 0, *liczba zlewisk*: 9148					
-	-	1788	-	1788	-
obraz: sufit_2, *skala*: 1, *liczba zlewisk*: 2231					
sRGB	Canny	28	2	30	60
sRGB	4-s	28	2	30	60
CIELab	Canny	28	3	31	58
CIELab	4-s	28	3	31	58
obraz: sufit_2, *skala*: 2, *liczba zlewisk*: 880					
sRGB	Canny	3	2	5	358
sRGB	4-s	3	3	6	298
CIELab	Canny	3	4	7	255
CIELab	4-s	3	4	7	255

Przenoszenie wyników segmentacji do oryginalnej rozdzielczości zajmuje zazwyczaj od 2 do 4 sekund. Czas ten zależy w największym stopniu od wykorzystanego filtru wykrywającego krawędzie. Filtr 4-sąsiedztwa generuje grubsze krawędzie niż filtr Canny'ego, co oznacza, że trzeba przypisać do klas więcej pikseli i stąd dłuższe czasy przenoszenia. Ponadto, wykorzystanie przestrzeni CIELab powoduje zazwyczaj wydłużenie czasu przenoszenia. W przypadku wszystkich obrazów jakość segmentacji w skali 1 była tylko nieznacznie niższa niż ta w skali 0. Segmentacja w skali 2, w przypadku części obrazów, była pozbawiona zbyt wielu szczegółów (rysunki 5.6c, 5.6i oraz 5.6o). Nie oznacza to, że wykonanie dwóch kroków transformaty falkowej jest nieuzasadnione, ponieważ często można uzyskać dobre wyniki (rysunki 5.6f oraz 5.6l). Bez względu na skalę zastosowanie filtru 4-sąsiedztwa powoduje uzyskanie bardziej zaokrąglonych krawędzi niż w przypadku filtru Canny'ego. Drugi z tych filtrów często powoduje powstanie efektu schodków na krawędziach obiektów. Algorytm przenoszenia wyników działa dobrze w przypadku obydwu przestrzeni barw: sRGB i CIELab. Jednakże, szczególnie w skali 2, widoczne jest zniekształcenie prostych krawędzi przy zastosowaniu filtru Canny'ego i obliczaniu odległości między pikselami w przestrzeni sRGB (rysunek 5.6o).

	skala 0	**skala 1**	**skala 2**
budynek_1, RGB, 4-sąsiedztwo (PCA, T: 0, P_z: 4, CLINK, śr.,war., n_k: 11)	(a)	(b)	(c)
kwiatek_3, RGB, Canny (PZS, T: 5, P_z: 10, UPGMA, śr., n_k: 9)	(d)	(e)	(f)
mewy_3, CIELab, 4-sąsiedztwo (PCA, T: 0, P_z: 1, CLINK, śr., war., n_k: 13)	(g)	(h)	(i)
mur_1, CIELab, 4-sąsiedztwo (PCA, T: 0, P_z: 4, UPGMA, śr., n_k: 2)	(j)	 (k)	 (l)
sufit_2, RGB, Canny (PCA, T: 0, P_z: 3, UPGMA, śr., war., n_k: 2)	 (m)	 (n)	 (o)

Rysunek 5.6: Wpływ skalowania obrazu na jakość końcowej segmentacji (pierwsza kolumna zawiera parametry przywracania pełnej rozdzielczości oraz, w nawiasach, parametry segmentacji).

Przedstawione wyniki potwierdzają, że zastosowanie transformaty falkowej może być użyte do przyspieszenia algorytmu klasteryzacji zlewisk. Osiągnięte przyspieszenia mogą przekraczać 300 w wypadku, gdy obraz w oryginalnej rozdzielczości jest w dużym stopniu nadmiernie posegmentowany. Optymalna skala segmentacji zależy od obrazu, jednak można stwierdzić, że skale większe niż 2 powodowałyby uzyskanie zbyt niskiej jakości. Podczas przenoszenia wyników do oryginalnej rozdzielczości przestrzeń CIELab powoduje mniejsze zniekształcenia krawędzi.

Rozdział 6

Adaptacyjny system wspomagający usuwanie nadsegmentacji

6.1 Wprowadzenie

Adaptacyjny system wspomagający usuwanie nadsegmentacji implementuje metodę redukcji nadsegmentacji przedstawioną w poprzednich rozdziałach. W kolejnych punktach przedstawiono wszystkie etapy segmentacji – od utworzenia bądź wczytania obrazu nadmiernie posegmentowanego do wyboru liczby klas i tworzenia ostatecznej segmentacji. System przewiduje również możliwość przyspieszenia procesu segmentacji dzięki wykorzystaniu transformaty falkowej. Ponadto możliwa jest ręczna korekcja segmentacji przez użytkownika systemu, w wypadku gdyby uzyskana segmentacja nie odpowiadała jego wymaganiom.

Oprócz funkcji ściśle związanych z transformacją wododziałową system oferuje: (1) możliwość konwersji przestrzeni barw, (2) eksportowanie składowych do obrazów monochromatycznych (separacja składowych), (3) zastępowanie wybranej składowej poprzez obraz monochromatyczny (import składowej), (4) ustawienie wybranej wartości we wszystkich pikselach składowej (umożliwia to wyeliminowanie wpływu składowej na wynik segmentacji), (5) filtry rozmywające, (6) filtry gradientowe, (7) możliwość dostosowania rozmiarów obrazu do wymagań transformaty falkowej.

6.2 Uzyskanie nadmiernie posegmentowanego obrazu

Opracowany system pozwala na wygenerowanie obrazu z nadsegmentacją na podstawie obrazu oryginalnego – czyli na przeprowadzenie całego procesu segmentacji obrazu od początku (wstępnym przetwarzaniem obrazu, obliczeniem obrazu gradientowego oraz wyznaczeniem transformacji wododziałowej). Pozwala również na przeprowadzenie redukcji nadsegmentacji za pomocą analizy skupień w obrazie, który został wcześniej poddany transformacji wododziałowej.

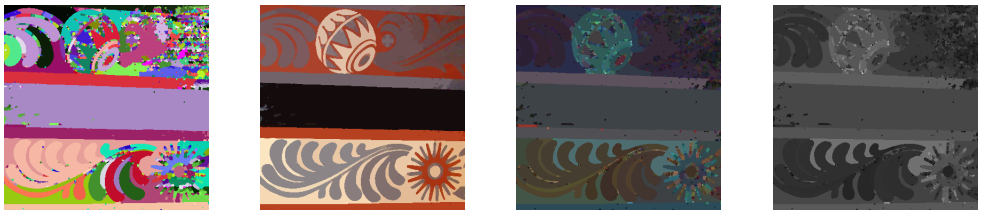
W przypadku pierwszego rozwiązania obraz jest przetwarzany zgodnie ze schematem przedstawionym na rysunku 3.1. Użytkownik systemu wybiera selektywny filtr rozmywający spośród filtrów przedstawionych w punkcie 3.2.1. Możliwa jest również rezygnacja z rozmywania obrazu. Gdy jednak wybrano jeden z filtrów, konieczne jest określenie ich parametrów (konduktancji, liczby iteracji oraz przyrostu czasu), które mają wpływ na siłę filtru oraz dokładność uzyskanego wyniku. Dodatkowo, w przypadku obrazów barwnych, użytkownik określa sposób obliczania obrazu gradientowego. Dostępne są dwie metody: *gradient-skp* (równanie 2.21) oraz *gradient-pca* (równanie 2.24). W przypadku obrazów monochromatycznych zawsze stosowany jest wzór 2.20. Następnie trzeba jeszcze określić dwa parametry: próg, który służy do usuwania niewielkich wartości gradientu, oraz poziom łączenia wykorzystywany przy łączeniu płytkich zlewisk. Możliwe jest wyłączenie progowania oraz zalewania płytkich zlewisk poprzez ustawienie odpowiednich wartości na 0. Gdy użytkownik zatwierdzi wybrane ustawienia, system przeprowadza transformację wododziałową oraz wstępną redukcję nadsegmentacji. Następnie prezentuje uzyskane wyniki i umożliwia dobór parametrów klasteryzacji.

Jak już wspomniano wcześniej system umożliwia również przeprowadzenie redukcji nadsegmentacji w obrazie wcześniej poddanym segmentacji. Oprócz obrazu oryginalnego, potrzebnego do obliczania wartości atrybutów zlewisk, system wymaga wczytania dodatkowego obrazu zawierającego nadmiernie posegmentowane obiekty. Powinien on mieć ten sam rozmiar co oryginał, a jego piksele powinny posiadać etykiety zlewisk, do których należą. Etykiety muszą być liczbami całkowitymi. Po wczytaniu wcześniej przeprowadzonej segmentacji system przechodzi bezpośrednio do etapu, w którym dokonuje się wyboru parametrów klasteryzacji.

6.3 Dobór parametrów klasteryzacji

Dobór parametrów klasteryzacji jest ważnym etapem. W zależności od dokonanego wyboru omawiany system może tworzyć segmentacje pozwalające na wydobycie z obrazu różnych jego cech. W pierwszej kolejności zostaną omówione istniejące możliwości wyboru ustawień.

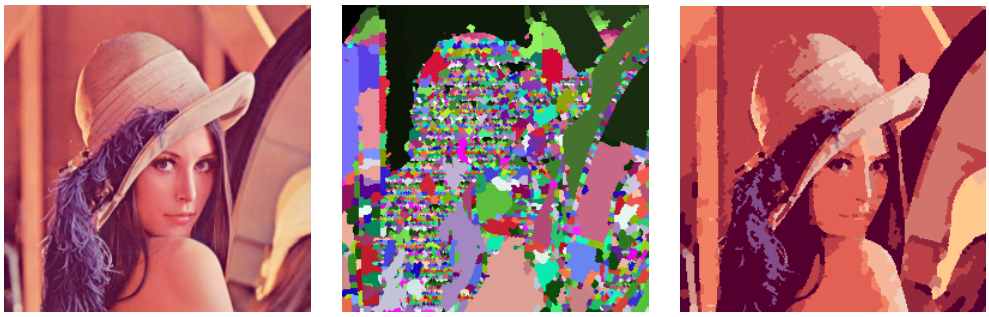
Po obliczeniu transformacji wododziałowej i przeprowadzeniu wstępnej redukcji nadsegmentacji system prezentuje uzyskaną segmentację. Zazwyczaj, mimo wstępnej redukcji, obraz jest w znacznym stopniu nadmiernie posegmentowany



(a) (b) (c) (d)

Rysunek 6.1: Różne sposoby przedstawiania obrazu nadmiernie posegmentowanego: (a) zlewiska o podobnych etykietach zostały pokolorowane za pomocą wyraźnie różniących się barw, (b) zlewiska reprezentowane przez swoje średnie, (c) zlewiska reprezentowane przez wartości odchylenia standardowego, (d) zlewiska reprezentowane przez wartości odchylenia standardowego obliczonego dla składowej niebieskiej obrazu RGB.

i zawiera od 1 do 4 tysięcy zlewisk. Istnieje możliwość oglądania tego wyniku w różnych postaciach. Pierwszą z nich jest obraz, w którym odcienie szarości odpowiadają poszczególnym etykietom. Bardziej czytelną wizualizacją wyników jest postać, w której kolejnym etykietom zlewisk przypisywane są wyraźnie różniące się od siebie barwy (przykład przedstawiono na rysunku 6.1a). Możliwe jest również wyświetlenie wartości wszystkich dostępnych atrybutów opisanych w punkcie 3.3.1. Przykłady dla średniej i odchylenia standardowego przedstawiono na rysunku 6.1b i 6.1c. Prezentacja wyników transformacji wododziałowej została połączona z wyborem parametrów analizy skupień. Dostępne są metody klasteryzacji, standaryzacji oraz miary odległości/podobieństwa opisane w dodatku A. System wymaga również utworzenia listy atrybutów. W trakcie jej tworzenia dostępne są wszystkie atrybuty przedstawione w punkcie 3.3.1. W przypadku obrazów barwnych są one dodawane grupami tzn. tworzonych jest tyle atrybutów, ile jest składowych w obrazie. Przykładowo podczas dodawania *średniej* dla obrazu RGB tworzone są: średnia składowej R, średnia składowej G oraz średnia składowej B. Tworzone w ten sposób grupy są numerowane na przykład po dodaniu średniej i wariancji do listy atrybutów wszystkie wartości średnie znajdują się w grupie 0, a wszystkie wariancje w grupie 1. Dzięki temu są one wspólnie standaryzowane w sposób opisany w punkcie 4.6. Warto dodać, iż użytkownik może zmienić zachowanie systemu i dodawać do listy atrybuty wyłącznie dla wybranych składowych (Przykład wartości odchylenia standardowego obliczonych tylko dla składowej niebieskiej przedstawiono na rysunku 6.1d). Może też dowolnie grupować atrybuty – na przykład łączyć w grupy atrybuty różnych rodzajów lub wymusić osobne traktowanie każdego z nich. W przypadku obrazów monochromatycznych atrybuty domyślnie nie są grupowane. Po wybraniu i zatwierdzeniu przez użytkownika ustawień analizy skupień system tworzy hierarchię podobieństwa zlewisk. Następnymi krokami są prezentacja wyników i dobór liczby klas



(a) (b) (c)

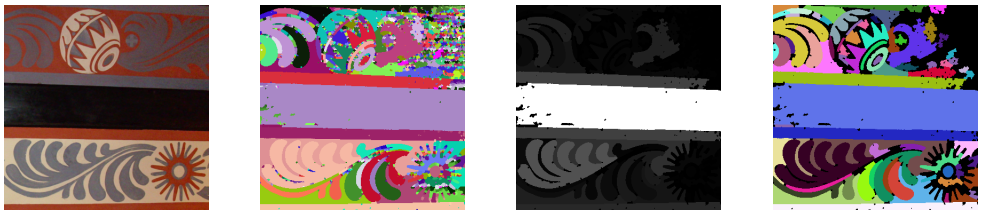
Rysunek 6.2: Segmentacja obrazu *Lena* zapisanego w przestrzeni *Lab*: (a) obraz oryginalny (*rozmiar:* 256 × 256, źródło: `http://www.ii.uni.wroc.pl/~anl/dyd/PO/examples1/`), (b) nadmiernie posegmentowany obraz (*liczba zlewisk:* 3078), (c) segmentacja (*gradient:* skp, *próg:* 1%, *poziom łączenia* 10%, *metoda:* CLINK, *współczynnik:* odl. euklidesowa, *standaryzacja:* brak, *atrybuty:* śr., *liczba klas:* 13)

opisane w kolejnych rozdziałach.

Dzięki sposobowi, w jaki zaimplementowano proponowane rozwiązanie, system wspomagający usuwanie nadsegmentacji może być bardzo szybko zaadaptowany do uzyskania określonego celu segmentacji. Przedstawione zostaną trzy różne przykłady: (1) segmentacja obrazu ze względu na różnice barw postrzegane przez człowieka, (2) segmentacja zaszumionego obszaru obrazu, (3) segmentacja obszarów z pominięciem intensywności barwy.

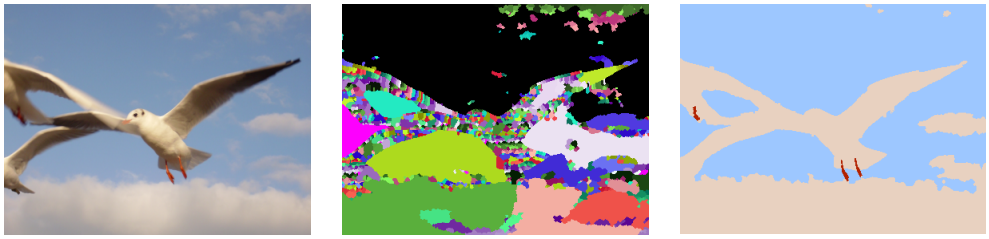
Pierwszy z przykładów znajduje się na rysunku 6.2 i przedstawia segmentację obrazu *Lena*. Obraz przetwarzany był w przestrzeni *CIELab*. W zestawie atrybutów umieszczono wyłącznie średnie zlewisk obliczone dla poszczególnych zlewisk. Ze względu na to, że przestrzeń CIELab jest percepcyjnie równomierna, do porównywania zlewisk wykorzystano zwykłą odległość euklidesową. Jak widać wynik segmentacji dobrze odzwierciedla różnice między barwami postrzegane przez człowieka, mimo że zlewiska zostały połączone w niewielką liczbę klas.

Druga przykładowa segmentacja została przedstawiona na rysunku 6.3. W obrazie oryginalnym *sufit_2* (rysunek 6.3a) niektóre obszary charakteryzują się wysokim poziomem szumu, który jest szczególne silny w prawym górnym rogu obrazu. W celu wysegmentowania zaszumionego obszaru na liście atrybutów umieszczono tylko rozmiar zlewiska. Wykorzystano tutaj cechę charakterystyczną transformacji wododziałowej stosowanej w połączeniu z filtrem gradientowym. Występowanie szumu powoduje pojawienie się większej liczby minimów lokalnych w obrazie gradientowym, a co za tym idzie, większej nadsegmentacji (rysunek 6.3b). Zlewiska powstające z powodu szumu są niewielkie. Do ich grupowania wykorzystano metodę pojedynczego wiązania (SLINK) – dzięki temu do regionu charakteryzującego się wysokim poziomem szumu były dołączane kolejne niewiel-



(a) (b) (c) (d)

Rysunek 6.3: Segmentacja zaszumionego obszaru na obrazie *sufit_2*: (a) obraz oryginalny, (b) zlewiska w nadmiernie posegmentowanym obrazie, (c) rozmiary poszczególnych zlewisk (jasne obszary odpowiadają dużym zlewiskom, ciemne obszary odpowiadają małym zlewiskom), (d) wysegmentowany zaszumiony obszar (zaznaczony na czarno) (*gradient:* skp, *próg:* 5%, *poziom łączenia* 10%, *metoda:* SLINK, *współczynnik:* odl. euklidesowa, *standaryzacja:* brak, *atrybuty:* rozm.)

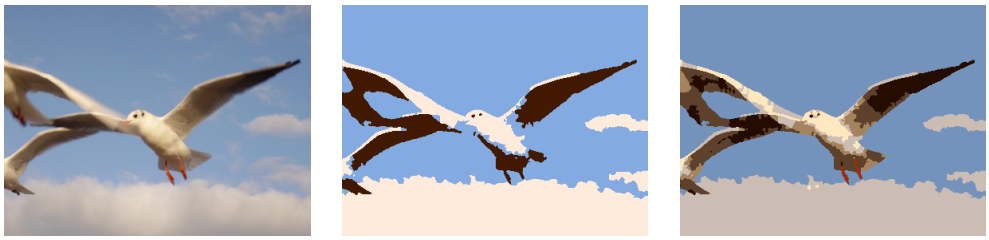


(a) (b) (c)

Rysunek 6.4: Segmentacja obrazu *mewy_3* bez uwzględniania jasności barwy, (a) obraz oryginalny (b) nadsegmentacja (1512 zlewisk) (c) segmentacja (*gradient:* pca, *próg:* 0%, *poziom łączenia* 1%, *metoda:* CLINK, *współczynnik:* kosinusowy, *standaryzacja:* brak, *atrybut:* średnia, *liczba klas:* 3)

kie zlewiska. Wynik segmentacji przedstawiono na rysunku 6.3d.

Trzeci przykład prezentuje możliwość uzyskania segmentacji obrazu, w której zlewiska grupowane są z pominięciem jasności barwy. Rysunek 6.4a przedstawia obraz testowy *mewy_3*. Na zdjęciu występują obszary o podobnym odcieniu lecz różniące się intensywnością (białe mewy i biało–szare chmury). Obraz przetwarzany był w przestrzeni RGB, w której jasność barwy nie jest bezpośrednio dostępna. Z tego względu jako miarę podobieństwa przyjęto współczynnik kosinusowy, ponieważ jest on niewrażliwy na skalowanie (mnożenie przez stałą) atrybutów. Ponieważ zestaw atrybutów składał się wyłącznie ze średnich (dla 3 składowych), zlewiska, których średnie różniły się wyłącznie jasnością, były uznawane za identyczne. Warto również dodać, że barwy różniące się nawet dwukrotnie nasyceniem (w przestrzeni HSV), charakteryzują się według współczynnika kosinusowego, wysokim stopniem podobieństwa. Uzyskana segmentacja znajduje się na rysunku 6.4c. Dla porównania przeprowadzono również drugą segmentację



(a) (b) (c)

Rysunek 6.5: Segmentacje obrazu *mewy_3*, (a) obraz oryginalny (b) segmentacja – *liczba klas:* 3, (c) segmentacja – *liczba klas:* 7, (wspólne parametry – *gradient:* pca, *próg:* 0%, *poziom łączenia* 1%, *metoda:* CLINK, *współczynnik:* odl. euklidesowa, *standaryzacja:* brak, *atrybut:* średnia)

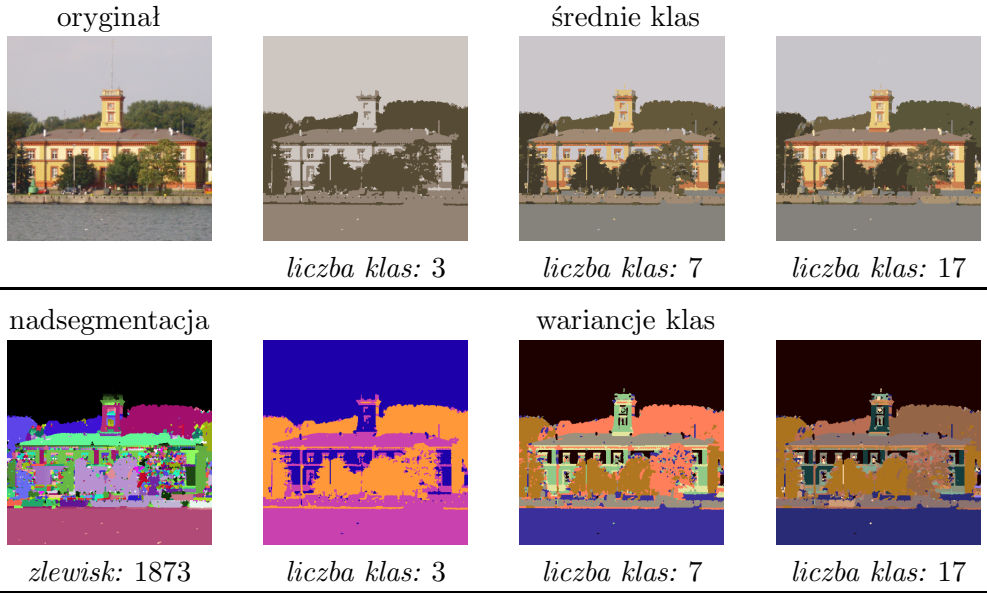
z zastosowaniem odległości euklidesowej. Wyniki przedstawiono na rysunku 6.5.

Efekty podobne do przedstawionych powyżej mogą być osiągnięte za pomocą innych metod. Na przykład, wynik podobny do tego z obrazu 6.4c można by uzyskać, stosując odpowiednią metodę segmentacji wykorzystującą przestrzeń HSV. Powyższe przykłady mają na celu pokazanie, jak łatwo można zmieniać efekty segmentacji uzyskiwane na pomocą omawianego systemu wspomagającego usuwanie nadsegmentacji.

6.4 Generowanie segmentacji o różnej liczbie klas

Po zakończeniu analizy skupień, system wspomagający usuwanie nadsegmentacji generuje na podstawie hierarchii podobieństwa tablice LUT według schematu przedstawionego na rysunku 3.10. Umożliwiają one szybkie generowanie segmentacji z określoną liczbą klas. Dzięki temu użytkownik systemu może w sposób interaktywny zmieniać liczbę klas i oceniać jakość segmentacji.

Rysunek 6.6 przedstawia segmentacje obrazu testowego *budynek_1*. Wszystkie segmentacje wygenerowano na podstawie tej samej hierarchii podobieństwa. W trakcie dokonywania wyboru liczby klas wszystkie zmiany parametrów są natychmiast odzwierciedlane w wynikach prezentowanych użytkownikowi. Bieżąca segmentacja może być przedstawiona, podobnie jak wartości atrybutów, w wielu postaciach. Domyślnym sposobem prezentacji jest monochromatyczny obraz, w którym różne odcienie szarości odpowiadają poszczególnym etykietom zlewisk. Także i w tym wypadku możliwa jest wizualizacja wartości wszystkich dostępnych atrybutów. Różnica polega na tym, że są one obliczane nie dla poszczególnych zlewisk, lecz dla klas w bieżącej segmentacji. Na rysunku 6.6 przedstawiono wartości średnich i wariancji klas w wygenerowanych segmentacjach. Obok etykiet lub wartości wybranego atrybutu system prezentuje równocześnie segmentację, w której klasom przyporządkowano różniące się barwy. Ułatwia to rozróżnianie



Rysunek 6.6: Segmentacje obrazu *budynek_1* zostały utworzone na podstawie tej samej hierarchii podobieństwa. (parametry – *gradient:* pca, *próg:* 0%, *poziom łączenia* 9%, *metoda:* CLINK, *współczynnik:* odl. euklidesowa, *standaryzacja:* brak, *atrybuty:* śr., war.)

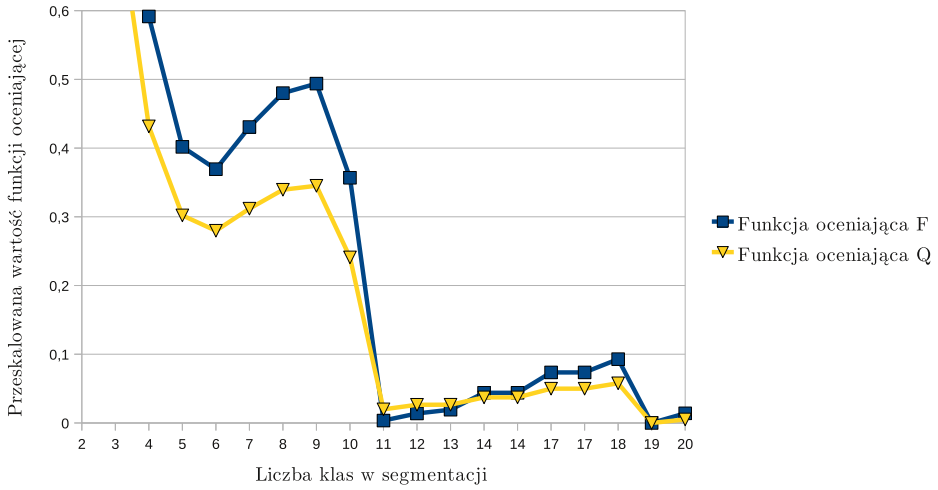
klas o podobnych etykietach czy wartościach wybranego atrybutu. W dowolnym momencie istnieje również możliwość zapisania bieżącej segmentacji.

6.5 Optymalizacja liczby klas

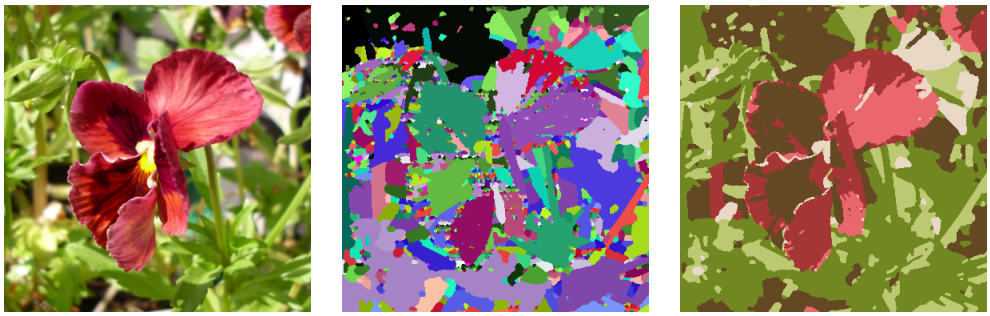
W trakcie wyboru liczby klas system wspomagający usuwanie nadsegmentacji umożliwia również obliczanie wartości funkcji oceny segmentacji. Dzięki temu użytkownik ma możliwość nie tylko jakościowej, ale także ilościowej oceny segmentacji. Wszystkie dostępne funkcje są opisane w rozdziale 4. Pozwalają one ocenić, jak zmiana liczby klas wpływa na stopień zgodności segmentacji i obrazu oryginalnego.

Funkcje, które mają wbudowany współczynnik kary pogarszający ocenę segmentacji ze zbyt dużą liczbą klas, mogą być wykorzystane do optymalizacji liczby klas. Rysunek 6.7 przedstawia wykres funkcji F (równanie 4.15) i Q (równanie 4.18) otrzymanych dla segmentacji obrazu *kwiatek_3* przy zmieniającej się liczbie klas (od 2 do 20). Wartości każdej z funkcji przeskalowano do przedziału $[0;1]$, aby ułatwić prezentację i porównanie funkcji. Funkcję F' (równanie 4.17) pominięto, ponieważ po przeskalowaniu jej kształt (w przypadku obrazu *kwiatek_3*) nie różnił się w widoczny sposób od funkcji F.

W przypadku funkcji F, F' i Q mniejsze wartości odpowiadają lepszym oce-



Rysunek 6.7: Wykres przedstawiający zależność wartości funkcji F oraz Q dla obrazu *kwiatek_3* (*gradient:* skp, *próg:* 5%, *poziom łączenia* 10%, *metoda:* Warda, *atrybuty:* śr., odch. std.). Wartości przeskalowano do przedziału $[0; 1]$. Na wykresie przedstawiono tylko wartości z zakresu $[0; 0,6]$, aby minima lokalne były lepiej widoczne.



obraz oryginalny *zlewisk:* 1965 *klas:* 6

Rysunek 6.8: Segmentacje obrazu *kwiatek_3* wybrane na podstawie wartości F i Q (*gradient:* skp, *próg:* 5%, *poziom łączenia* 10%, *metoda:* Warda, *atrybuty:* śr., odch. std.)

obraz oryginalny | *klas:* 11 | *klas:* 19

Rysunek 6.9: Kolejne segmentacje obrazu *kwiatek_3* wybrane na podstawie wartości F i Q (*gradient:* skp, *próg:* 5%, *poziom łączenia* 10%, *metoda:* Warda, *atrybuty:* śr., odch. std.)

nom. Jak widać na rysunku 6.7 funkcje te mogą posiadać wiele minimów lokalnych. Znalezienie minimum globalnego może być długotrwałe, ponieważ wymagałoby wygenerowania wszystkich możliwych segmentacji (ich liczba może wyrażać się w tysiącach) i obliczenia dla nich wartości funkcji oceny. Dzięki temu, że celem zastosowania omawianego systemu jest redukcja nadsegmentacji możliwe jest znaczące ograniczenie przedziału, w którym poszukiwane są minima. Rysunki 6.8 oraz 6.9 przedstawiają segmentacje obrazu *kwiatek_3* odpowiadające minimom lokalnym funkcji oceny.

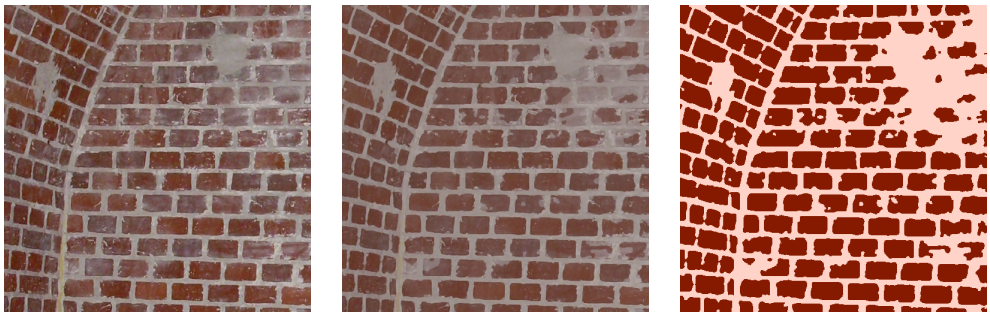
Oprócz obliczania wartości funkcji oceny dla wygenerowanej segmentacji, system wspomagający usuwanie nadsegmentacji umożliwia, po wybraniu funkcji oceny, określenie zakresu poszukiwań optymalnej liczby klas. Następnie automatycznie generuje segmentacje, oblicza ich oceny i na tej podstawie wybiera optymalną liczbę klas. W przypadku funkcji F, F' i Q system poszukuje minimum wybranej funkcji w określonym zakresie. Ze względu na to, że pozostałe funkcje określają stopień w jakim segmentacja odpowiada obrazowi oryginalnemu sposób poszukiwania optymalnej liczby klas jest inny. Nowe segmentacje generowane są tak długo, jak długo poprawa funkcji oceny przekracza pewien zadany przez użytkownika próg. Gdy poprawa wartości funkcji jest zbyt mała, system uznaje, że dalsze zwiększanie liczby klas jest bezcelowe. Użytkownik ma możliwość ręcznej korekty wartości wybranej automatycznie.

6.6 Przyspieszanie za pomocą falek

W rozdziale 5 przedstawiono sposób poprawy wydajności przedstawionego w niniejszej pracy rozwiązania. Polega ono na (1) zmniejszeniu rozdzielczości obrazu za pomocą transformaty falkowej, (2) przeprowadzeniu segmentacji mniejszej wersji obrazu (według schematu przedstawionego na rysunku 5.1) i (3) przeniesieniu

obraz oryginalny (512 × 512) | 1 krok transformaty falkowej | 2 kroki transformaty falkowej

Rysunek 6.10: Ilustracja interaktywnego wyboru liczby kroków transformaty falkowej.



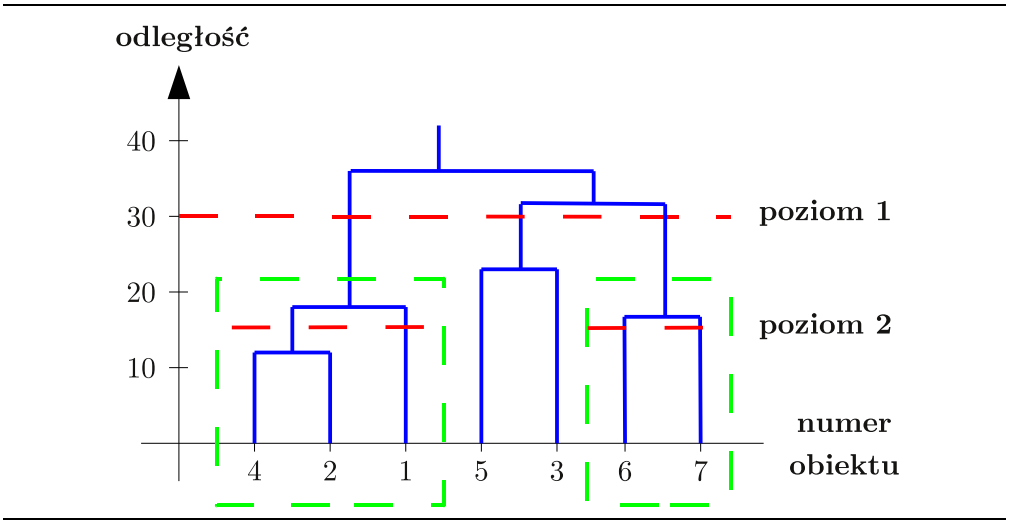
obraz oryginalny | 50% oryginału + 50% segmentacji | segmentacja (2 klasy)

Rysunek 6.11: Ilustracja interaktywnego porównania obrazu oryginalnego z wynikiem segmentacji. Segmentacja, w której klasy reprezentowane są przez swoje średnie, nakładana jest na obraz oryginalny w określonych proporcjach.

wyników do oryginalnej rozdzielczości (według schematu przedstawionego na rysunku 5.4). W stosunku do procesu segmentacji opisanego w punktach 6.2–6.5 występują dwie różnice.

Pierwszą z nich jest wybór skali, w której przeprowadzana jest segmentacja. Wybór musi być dokonany przed przeprowadzeniem transformacji wododziałowej. Użytkownik systemu wybiera skalę. Transformata falkowa jest obliczana na bieżąco a jej wyniki natychmiast są prezentowane użytkownikowi. Przykładowe wyniki dwóch pierwszych kroków transformaty falkowej przedstawia rysunek 6.10.

Drugą różnicą w stosunku do procesu segmentacji przedstawionego w punktach 6.2–6.5 jest przeniesienie wyników do rozdzielczości obrazu oryginalnego. Użytkownik wybiera niezbędne parametry: (1) filtr stosowany do wykrywania krawędzi zlewisk, (2) przestrzeń, w której obliczane są odległości między barwami, (3) normę stosowaną do obliczania tych odległości. Następnie system tworzy



Rysunek 6.12: Wybór różnych poziomów przecięcia hierarchii podobieństwa dla fragmentów obrazu umożliwia dzielenie klas na mniejsze fragmenty.

segmentację w pełnej rozdzielczości. Na koniec system umożliwia użytkownikowi porównanie otrzymanej segmentacji z oryginalnym obrazem. Porównanie odbywa się na zasadzie mieszania obrazów w określonych proporcjach, które mogą być płynnie zmieniane przez użytkownika i powodować przejście od obrazu oryginalnego do segmentacji. Rysunek 6.11 przedstawia obraz oryginalny *mur_1*, jego segmentację i przenikające się obrazy.

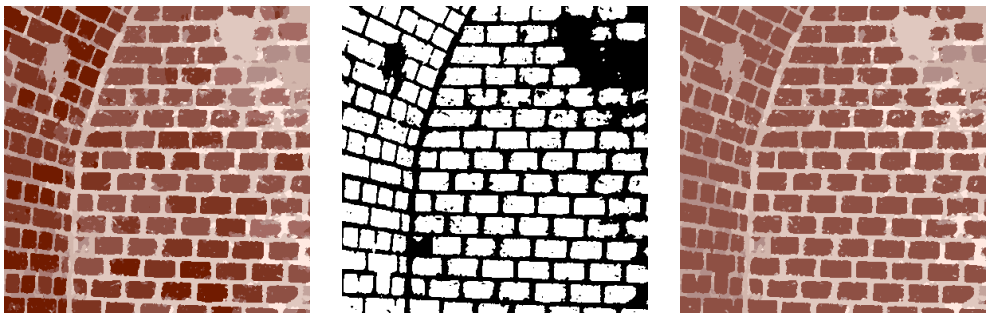
6.7 Ręczna korekcja segmentacji

Wyniki uzyskane za pomocą analizy skupień mogą być, w niektórych przypadkach, poprawione poprzez ręczną korekcję segmentacji. Dotyczy to sytuacji dwóch rodzajów. W sytuacji pierwszego rodzaju otrzymana segmentacja jest prawidłowa w stosunku do większości obiektów, jednak niektóre z nich zostały niepotrzebnie połączone. W sytuacji drugiego rodzaju niektóre obiekty przedstawione na obrazie są w nieznacznym stopniu nadmiernie posegmentowane. Opisywany system przewiduje możliwość korekcji segmentacji w przypadku każdego ze wspomnianych problemów.

Idea podziału niepotrzebnie połączonych obiektów polega na umożliwieniu użytkownikowi wyboru osobnych progów dla wybranych obszarów obrazu. Rysunek 6.12 przedstawia przykładową hierarchię podobieństwa, dla której użytkownik wybrał *poziom 1*. Zlewiska 1, 2, 4 oraz 6 i 7 zostały niepotrzebnie ze sobą połączone i stanowią 2 klasy w bieżącej segmentacji. Po wybraniu obszarów należących do tych klas użytkownik wybrał *poziom 2*. W ten sposób zlewiska 6 i 7 oraz 1 znalazły się w osobnych klasach. Zachowane zostało połączenie zlewisk 2 i 4.

(a) (b) (c) (d)

Rysunek 6.13: Podział klasy na mniejsze części w segmentacji obrazu *budynek_1*: (a) segmentacja zawierająca 7 klas (niebo i część żółtej elewacji budynku zostały błędnie połączone w jedną klasę), (b) klasa, którą należy podzielić (biały obszar), (c) segmentacja zawierająca 17 klas (niebo i elewacja nie są połączone), (d) ostateczna segmentacja – połączenie obrazów (a) i (c).



(a) (b) (c)

Rysunek 6.14: Łączenie klas w segmentacji obrazu *mur_1*: (a) segmentacja zawierająca 10 klas, (b) klasy do połączenia (biały obszar), (c) ostateczna segmentacja po połączeniu klas.

Rysunek 6.13 przedstawia przykład zastosowania powyższego rozwiązania na przykładzie rzeczywistej segmentacji. Na rysunku 6.13a widać, że nadsegmentacja została znacząco zredukowana. Jednakże dwa obiekty zostały błędnie połączone – niebo oraz część żółtej elewacji budynku znalazły się w jednej klasie. Ta klasa została wybrana jako przeznaczona do podziału (biały obszar na rysunku 6.13b). Następnie ustawiono próg, przy którym wspomniane obiekty nie znajdują się w tej samej klasie. Tworzona jest tymczasowa segmentacja z nowym podziałem klas (rysunek 6.13c). Segmentacje 6.13a i 6.13c są na siebie nakładane – wybrany obszar w segmentacji początkowej jest zastępowany fragmentem segmentacji tymczasowej (rysunek 6.13d).

System umożliwia wielokrotne powtarzanie przedstawionego powyżej schematu postępowania. Użytkownik może zaznaczyć dowolne klasy w bieżącej segmentacji, wybrać nowy próg podziału hierarchii podobieństwa i w ten sposób zmienić połączenia zlewisk. Warto zaznaczyć, że w faktycznej implementacji segmentacje

są generowane oraz łączone na bieżąco, co umożliwia interaktywny dobór nowego progu i ułatwia ocenę otrzymywanych wyników.

Jak już wspomniano korekcja segmentacji poprzez łączenie klas jest przydatna w sytuacji, w której niektóre obiekty są nadmiernie posegmentowane i nie jest możliwe wybranie takiego progu podziału hierarchii podobieństwa, który spowodowałby ich połączenie bez wprowadzania innych błędów w segmentacji. Łączenie jest prostym procesem. Użytkownik systemu wybiera klasy, które należy połączyć i po zatwierdzeniu wyboru otrzymuje skorygowaną segmentację. W tym wypadku również efekty wprowadzanych zmian są na bieżąco prezentowane użytkownikowi. Rysunek 6.14 przedstawia przykład łączenia klas.

Rozdział 7

Podsumowanie

Realizując cel niniejszej pracy, autor, w pierwszej kolejności zaproponował adaptacyjną metodę redukcji nadsegmentacji w obrazach poddanych transformacji wododziałowej, a następnie zaimplementował system pozwalający na wykorzystanie tej metody w praktyce. System ten umożliwia przeprowadzenie segmentacji od początku (łącznie z przetwarzaniem wstępnym i wyznaczeniem transformacji wododziałowej), a także usuwanie nadsegmentacji w obrazie, dla którego transformację wododziałową obliczono wcześniej. Możliwe jest, poprzez odpowiedni dobór parametrów, dostosowanie sposobu klasteryzacji zlewisk, i wpływanie w ten sposób na kryteria segmentacji. Wyniki klasteryzacji są przetwarzane w sposób interaktywny. Użytkownik ma możliwość samodzielnego wyboru liczby klas w końcowej segmentacji, a także automatycznego wyboru opartego na wartościach funkcji oceniających. Ten wybór może zostać później skorygowany. W wypadku, gdyby otrzymana w wyniku klasteryzacji hierarchia podobieństwa zlewisk, nie pozwalała na uzyskanie satysfakcjonującej segmentacji, system przewiduje możliwość ręcznego łączenia kilku klas w jedną lub dzielenia błędnie połączonych klas. Gdy poddawane segmentacji obrazy są w wysokiej rozdzielczości proces można przyspieszyć dzięki transformacie falkowej. Za jej pomocą segmentowany obraz jest pomniejszany, przeprowadzana jest segmentacja (wraz z przeprowadzeniem redukcji nadsegmentacji), a następnie jej wyniki przenoszone są do pełnej rozdzielczości obrazu.

W celu potwierdzenia słuszności postawionych tez przeprowadzono serię eksperymentów. Pierwszy z nich pokazał, że wybrane istniejące metody redukcji nadsegmentacji, nawet zastosowane razem, nie dają zadowalających wyników. Użycie selektywnego filtru rozmywającego okazało się efektywne, ale wyłącznie dla obrazu dość silnie zaszumionego. Progowanie oraz zalewanie płytkich zlewisk pozwala nawet na kilkukrotne zmniejszenie liczby regionów w obrazie. Trzeba jednak zaznaczyć, że wspomniane metody pozwalają na zmniejszenie liczby regio-

nów, w segmentacjach obrazów testowych, do około 2000 z początkowych 12000 – 20000. Próba silniejszej redukcji nadsegmentacji kończy się błędnym łączeniem regionów należących do różnych obiektów przy dość dużej liczbie regionów (około 500). Powyższa próba pokazała konieczność zastosowania kolejnego etapu redukcji nadsegmentacji. Autor niniejszej pracy zaproponował podejście, w którym wykorzystuje się wspomniane rozwiązania do wstępnej redukcji nadsegmentacji, a następnie za pomocą analizy skupień łączy je w ostateczne klasy. Pozwoliło to na dalszą redukcję nadsegmentacji. Dla obrazów testowych liczbę klas zmniejszono do poziomu 10–20. Takie wartości były bliższe rzeczywistej liczbie obiektów przedstawionych na obrazach.

Drugi eksperyment miał na celu stwierdzenie, które parametry analizy skupień zapewniają najlepsze wyniki klasteryzacji zlewisk. Próbę przeprowadzono na czterech monochromatycznych obrazach: CT, MRI T1, MRI T2 oraz MRI PD. Uwzględniono w niej wszystkie metody klasteryzacji, miary podobieństwa/odległości oraz metody standaryzacji opisane w dodatku A. Atrybuty zlewisk połączono w dwanaście różnych zestawów. Powyższe parametry łączono na wszystkie możliwe sposoby i z ich zastosowaniem wykonywano klasteryzację zlewisk w czterech testowych obrazach. Dla każdej uzyskanej hierarchii podobieństwa wygenerowano segmentacje zawierające różną liczbę klas. W ten sposób powstało około 58000 segmentacji. Następnie obliczano dla nich wartości funkcji oceny segmentacji. Na ich podstawie wybrano parametry, które najczęściej umożliwiały uzyskanie dobrych wyników. Okazało się, że optymalne parametry zależą od typu segmentowanego obrazu. Przykładowo w przypadku obrazów MRI PD najlepsza okazała się metoda minimalnej wariancji Warda, w przypadku MRI T1 oraz MRI T2 metoda Warda oraz metoda pełnego wiązania (CLINK), natomiast dla obrazów CT najlepsze są metody Warda oraz średniego wiązania (UPGMA). Warto też zauważyć, że większość najlepszych zestawów atrybutów zawierała więcej niż jeden atrybut.

Trzecia próba dotyczyła obrazów barwnych. Została przeprowadzona na pięciu barwnych obrazach naturalnych. Tak jak poprzednia, dotyczyła doboru optymalnych parametrów. Jej przebieg był podobny. Uwzględniono w niej takie same zestawy parametrów, jak w przypadku obrazów monochromatycznych. Jedyną różnicą było to, że wektory charakteryzujące poszczególne zlewiska – miały więcej elementów. Wartości atrybutów obliczano osobno dla kolejnych składowych obrazu. Ze względu na większą liczbę obrazów testowych oraz ze względu na fakt, że na podstawie każdej hierarchii podobieństwa tworzono większą liczbę segmentacji w sumie uzyskano ponad 125000 posegmentowanych obrazów. Otrzymane wyniki oceniano za pomocą funkcji zastosowanych w poprzednim eksperymencie (funkcje zostały dostosowane do obrazów barwnych), a także za pomocą funkcji przeznaczonych wyłącznie do oceny segmentacji obrazów barwnych. Analiza uzyskanych wyników pokazała, że we wszystkich wypadkach uzyskano bardzo do-

bre wyniki klasteryzacji zlewisk za pomocą metody Warda i zestawu atrybutów złożonego ze średniej i wariancji zlewiska.

Doboru parametrów klasteryzacji dla różnych typów obrazów dokonano ze względu na funkcje oceny jakości segmentacji. Często są to funkcje mierzące błąd popełniany przy przybliżaniu obrazu za pomocą jego segmentacji. Uzyskanie dobrej aproksymacji obrazu nie zawsze jest celem segmentacji. Dlatego zaprezentowano w pracy możliwość dostosowania opracowanego systemu do mniej typowych zadań np. segmentacji zaszumionych obszarów obrazu.

W czwartym eksperymencie sprawdzona została możliwość przyspieszenia procesu segmentacji dzięki zastosowaniu transformaty falkowej. Segmentacji poddawano pięć barwnych obrazów testowych. Każdy obraz był przetwarzany w trzech różnych skalach – porównano czasy i jakość segmentacji obrazów w oryginalnej rozdzielczości, obrazów o dwukrotnie mniejszej wysokości i szerokości oraz obrazów czterokrotnie pomniejszonych. Po segmentacji uzyskane klasy były przenoszone do oryginalnej rozdzielczości obrazu. Uzyskane wyniki pokazały, że dwukrotne zmniejszenie liniowych rozmiarów obrazu pozwala na 10–60 krotne przyspieszenie procesu segmentacji, po wliczeniu czasu potrzebnego na przenoszenie wyników do pełnej rozdzielczości. Czterokrotne zmniejszenie tych rozmiarów umożliwia uzyskanie przyspieszeń sięgających 300. Jednak stosowanie tej skali powoduje widoczny spadek jakości segmentacji. Warto również wspomnieć, że zastosowanie transformaty falkowej do budowy piramidy obrazów powoduje dodatkową redukcję nadsegmentacji. W przypadku większości obrazów testowych liczba regionów w nadmiernie posegmentowanym obrazie spadała kilkukrotnie wraz z dwukrotnym zmniejszeniem liniowych rozmiarów obrazu.

Wyniki powyższych eksperymentów pozwalają na wysnucie następujących wniosków:

1. Analiza skupień umożliwia poprawę efektów redukcji nadsegmentacji w porównaniu z efektem uzyskanym w wyniku selektywnego rozmywania obrazu, progowania obrazu gradientowego i zalewania płytkich zlewisk.

2. Zastosowanie transformaty falkowej pozwala na uzyskanie wyraźnego przyspieszenia procesu segmentacji. Dodatkowo powoduje zmniejszenie nadsegmentacji.

3. Poprzez odpowiedni dobór parametrów klasteryzacji system redukujący nadsegmentację można dostosować do różnych typów obrazów i różnych celów segmentacji.

Przeprowadzone, w ramach niniejszej pracy, badania pozwoliły na potwierdzenie słuszności postawionych tez i realizację celu pracy oraz pokazały, że przedstawione rozwiązania mogą być zastosowane do budowy metod segmentacji obrazu, które sprawdzą się w wielu zastosowaniach.

Bibliografia

[1] A. Achuthan, M. Rajeswari, D. Ramachandram. Detection of secondary structure in medical images using wavelet and watershed, 2006.

[2] P. S. U. Adiga, B. B. Chaudhuri. An efficient method based on watershed and rule-based merging for segmentation of 3-D histo-pathological images. *Pattern Recognition*, 34(7):1449–1458, 2001.

[3] Alaknanda, R. Anand, P. Kumar. Flaw detection in radiographic weldment images using morphological watershed segmentation technique. *NDT&E International*, 42:2–9, 2009.

[4] M. Andrade, G. Bertrand, A. Araújo. An attribute-based image segmentation method. *Materials Research*, 2(3):145–151, 1999.

[5] J. Angulo, J. Serra. Color segmentation by ordered mergings. *ICIP (2)*, strony 125–128, 2003.

[6] M. Baccar, L. Gee, R. Gonzalez, M. Abidi. Segmentation of range images via data fusion and morphological watersheds. *Pattern Recognition*, 29(10):1673–1687, 1996.

[7] F. Barré, J. Lopez. Watershed lines and catchment basins: a new 3d-motif method. *International Journal of Machine Tools and Manufacture*, 40:1171–1184, 2000.

[8] S. Bermejo, J. Cabestany. Oriented principal component analysis for large margin classifiers. *Neural Networks*, 14:1447–1461, 2001.

[9] S. Beucher. Extrema of grey-tone functions and mathematical morphology. Raport instytutowy N-793, Ecole des Mines de Paris, 1983.

[10] S. Beucher. The watershed transformation applied to image segmentation. *Scanning Microscopy International*, 6:299–314, 1992.

[11] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. J. Serra, P. Soille, redaktorzy, *Mathematical Morphology and its Applications to Image Processing*, strony 69–76. Kluwer Academic Publishers, 1994.

[12] S. Beucher. Applications of mathematical morphology in material sciences: a review of recent developments. *Proceedings MC95*, 1995.

[13] S. Beucher, M. Bilodeau. Road segmentation and obstacle detection by a fast watershed transformation. *Intelligent Vehicles '94 Symposium, Proceedings of the*, strony 296–301, 1994.

[14] S. Beucher, M. Bilodeau, X. Yu. Road segmentation by watersheds algorithms. *Proc. of PROMETHEUS workshop, Sophia-Antipolis*, 1990.

[15] S. Beucher, C. Lantuejoul. Use of watersheds in contour detection. *Real-time Edge and Motion Detection/Estimation*, International Workshop on Image Processing, 1979.

[16] S. Beucher, R. Peyrard, M. Bilodeau, M. Gauthier. Road monitoring and obstacle detection system by image analysis and mathematical morphology. *Proceedings 5th EAEC International Congress*, 1995.

[17] S. Beucher, X. Yu. Road recognition in complex traffic situations. *7th IFAC/IFORS Simposium on Transportation Systems: Theory and Application of Advanced Technology*, strony 413–418, 1994.

[18] J. T. Białasiewicz. *Falki i aproksymacje*. WNT, 2004.

[19] A. Bieniek, H. Burkhardt, H. Marschner, M. Noolle, G. Schreiber. A parallel watershed algorithm. *SCIA*, 1997.

[20] A. Bieniek, A. Moga. A connected component approach to the watershed segmentation. *In Mathematical Morphology and its Applications to Image and Signal Processing*, strony 215–222. Kluwer Academic Publishers, 1998.

[21] A. Bleau, L. J. Leon. Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, 77(3):317–370, 2000.

[22] N. Borel, A. Tremeau. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(10):1799–1800, 1997.

[23] M. Borsotti, P. Campadelli, R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19:741–747, 1998.

[24] G. Bueno, O. Musse, F. Heitz, J. P. Armspach. Three-dimensional segmentation of anatomical structures in mr images on large data bases. *Magnetic Resonance Imaging*, 19:73–88, 2001.

[25] J. E. Cates, R. T. Whitaker, M. Greg. Case study: an evaluation of user-assisted hierarchical watershed segmentation. *Medical Image Analysis*, 9:566–578, 2005.

[26] D. Chopp. Computing minimal surfaces via level set curvature flows. *Journal of Comparative Physics*, 106(1):77–91, 1993.

[27] L. D. Cohen. On active contour models and balloons. *Computer Vision Graphics and Image Processing: Image Understanding*, 53(2):211–218, 1991.

[28] J. Cousty, M. Couprie, L. Najman, G. Bertrand. Weighted fusion graphs: Merging properties and watersheds. *Discrete Applied Mathematics*, 156(15):3011–3027, 2008.

[29] A. Cumani. Edge detection in multispectral images. *CVGIP: Graphical Models and Image Processing*, 53(1):40–51, 1991.

[30] I. Dagher, K. E. Tom. Waterballoons: A hybrid watershed balloon snake segmentation. *Image and Vision Computing*, 26(7):905–912, 2008.

[31] S. Eom, S. Chang, B.-H. Ahn. Watershed-based region merging using conflicting regions. *ICIP (2)*, strony 781–784, 2002.

[32] B. S. Everitt, S. Landau, M. Leese. *Cluster Analysis*. Arnold, wydanie czwarte, 2001.

[33] M. Frucci, G. Ramella, G. S. di Baja. Using resolution pyramids for watershed image segmentation. *Image and Vision Computing*, 25(6):1021–1031, 2007.

[34] H. Gao, W. S. Lin, P. Xue, W. C. Siu. Marker-based image segmentation relying on disjoint set union. *SP:IC*, 21(2):100–112, 2006.

[35] G. Gerig, O. Kübler, R. Kikinis, F. A. Jolesz. Nonlinear anisotropic filtering if mri data. *IEEE Transactions on Medical Imaging*, 11(2):221–232, 1992.

[36] R. C. Gonzalez, R. E. Woods. *Digital Image Processing*. Pearson Education International, wydanie trzecie, 2008.

[37] V. Grau, M. A. Raya, C. Monserrat, M. C. Juan, L. Martí-Bonmatí. Hierarchical image segmentation using a correspondence with a tree model. *Pattern Recognition*, 37(1):47–59, 2004.

[38] M. Grimaud. New measure of contrast: dynamics. P. Gader, E. Dougherty, J. Serra, redaktorzy, *Image algebra and morphological image processing III*, wolumen SPIE-1769, strony 292–305, 1992.

[39] A. Hanbury, J. Serra. *Colour Image Analysis in 3D-Polar Coordinates*, wolumen 2781/2003 serii *Lecture Notes in Computer Science*, strony 124–131. Springer, 2003.

[40] A. G. Hanbury, J. Serra. Morphological operators on the unit circle. *IEEE Trans. Image Processing*, 10(12):1842–1850, 2001.

[41] R. M. Haralick, L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.

[42] K. Haris, S. Efstratiadis, N. Maglaveras. Hierarchical image segmentation based on contour dynamics. strony 54–57, 2001.

[43] K. Haris, S. N. Estradiadis, N. Maglaveras, A. K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, 7(12):1684–1699, 1998.

[44] F.-Y. Hsieh, C.-C. Han, N.-S. Wu, T. C. Chuang, K.-C. Fan. A novel approach to the detection of small objects with low contrast. *Signal Processing*, 86(1):71–83, 2006.

[45] L. Ibanez, W. Schroeder, L. Ng, J. Cates, i in. *The ITK Software Guide*. Kitware Inc., wydanie drugie, 2005.

[46] İsmail Avcibaş. *Image Quality Statistics and their Use in Steganalysis and Compression.* Praca doktorska, Uniwersytet Boğaziçi, 2001.

[47] K. Jiang, Q.-M. Liao, S.-Y. Dai. A novel white blood cell segmentation scheme using scale-space filtering and watershed clustering. *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, strony 2820–2825, 2003.

[48] C. R. Jung. Combining wavelets and watersheds for robust multiscale image segmentation. *Image and Vision Computing*, 25:24–33, 2007.

[49] C. R. Jung. Unsupervised multiscale segmentation of color images. *Pattern Recognition Letters*, 28(4):523–533, 2007.

[50] M. Kazanov. Modification of watershed transformation for images, containing small objects. *ICPR*, strony I: 612–615, 2004.

[51] M. Kazanov. A new color image segmentation algorithm based on watershed transformation. *ICPR*, strony II: 590–593, 2004.

[52] J.-B. Kim, H.-J. Kim. Multiresolution-based watersheds for efficient image segmentation. *Pattern Recognition Letters*, 24(1-3):473–488, 2003.

[53] M. Köppen, J. Ruiz-del-Solar, P. Soille. Texture segmentation by biologically-inspired use of neural networks and mathematical morphology. M. Heiss, redaktor, *NC'98, International ICSC/IFAC Symposium on Neural Computation*, strony 267–272. ICSC Academic Press, 1998.

[54] G. A. Korn, T. M. Korn. *Matematyka - dla pracowników naukowych i inżynierów*, wolumen 1. PWN, wydanie drugie, 1983.

[55] M. Krzysztofiak, D. Urbanek. *Metody statystyczne.* PWN, wydanie czwarte, 1981.

[56] W.-F. Kuo, C.-Y. Lin, Y.-N. Sun. Brain MR images segmentation using statistical ratio: Mapping between watershed and competitive hopfield clustering network algorithms. *Computer Methods and Programs in Biomedicine*, 91(3):191–198, 2008.

[57] M. M. Letteboer, O. F. Olsen, E. B. Dam, P. W. Willems, M. A. Viergever, W. J. Niessen. Segmentation of tumors in magnetic resonance brain images using an interactive multiscale watershed algorithm. *Academic Radiology*, 11(10), 2004.

[58] O. Lezoray, H. Cardot. Bayesian marker extraction for color watershed in segmenting microscopic images. *ICPR*, strony I: 739–742, 2002.

[59] O. Lezoray, H. Cardot. Histogram and watershed based segmentation of color images, 2002.

[60] O. Lezoray, A. Elmoataz, H. Cardot, M. Revenu. A color morphological segmentation. *In Proceedings of International Conference on Color in Graphics and Image Processing*, strony 170–175, 2000.

[61] H. Liu, Z. Chen, X. Chen, Y. Chen. Multiresolution medical image segmentation based on wavelet transform. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, strony 3418–3421, 2005.

[62] J. Liu, Y.-H. Yang. Multiresolution color image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):689–700, 1994.

[63] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997.

[64] R. Malladi, J. A. Sethian. Image processing via level set curvature flow. *Proc. Natl. Acad. of Sci.*, 92(15):7046–7050, 1995.

[65] N. Malpica, J. E. Ortuño, A. Santos. A multichannel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters*, 24:1545–1554, 2003.

[66] J. Mańdziuk. *Sieci neuronowe typu Hopfielda - Teoria i przykłady zastosowań.* Problemy współczesnej nauki - Teoria i zastosowania. Akademicka Oficyna Wydawnicza EXIT, 2000.

[67] D. Martin, C. Fowlkes, D. Tal, J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proc. 8th Int'l Conf. Computer Vision*, wolumen 2, strony 416–423, July 2001.

[68] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994.

[69] F. Meyer. The dynamics of minima and contours. *Mathematical Morphology and its Applications to Image and Signal Processing (ISMM'96)*, 1996.

[70] A. N. Moga, B. Cramariuc, M. Gabbouj. Parallel watershed transformation algorithms for image segmentation. *Parallel Computing*, 24(14):1981–2001, 1998.

[71] A. N. Moga, M. Gabbouj. Parallel marker-based image segmentation with watershed transformation. *Journal of Parallel and Distributed Computing*, 51(1):27–45, 1998.

[72] M. Naemura, A. Fukuda, Y. Mizutani, Y. Izumi, Y. Tanaka, K. Enami. Morphological segmentation of sport scenes using color information. *IEEE Transactions on Broadcasting*, 46(3), 2000.

[73] H. T. Nguyen, M. Worring, R. van den Boomgaard. Watersnakes Energy-driven watershed segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3), 2003.

[74] M. Nieniewski. *Segmentacja obrazów cyfrowych - Metody segmentacji wododziałowej.* Problemy współczesnej nauki - Teoria i zastosowania. Akademicka Oficyna Wydawnicza EXIT, 2005.

[75] B. Obara. A new algorithm using image colour system transformation for rock grain segmentation. *Mineralogy and Petrology*, 91:271–285, 2007.

[76] H. Palus, T. Kotyczka. Evaluation of colour image segmentation results. Raport instytutowy 15, Institut fuer Informatik Friedrich-Alexander-Universitaet Erlangen-Nuernberg, 2001.

[77] T. Pałys. Zastosowanie metody grupowania sekwencji czasowych do rozpoznawania mowy na podstawie Ukrytych Modeli Markowa. *VI Międzynarodowa Konferencja Naukowa Zarządzanie Kryzysowe*, 2008.

[78] C. Pan, C. Zheng, H.-J. Wang. Robust color image segmentation based on mean shift and marker-controlled watershed algorithm. *Proceedings of Second International Conference on Machine Learning and Cybernetics*, strony 2752–2756, 2003.

[79] D. Paulus, T. Greiner, C. L. Knüvener. Watershed transformation of time series of medical thermal images, 1995.

[80] P. Perona, J. Malik. Scale-space and edge detection using anisotropic diffusion. *Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[81] M. Piłatowska. *Repetytorium ze statystyki*. Wydawnictwo Naukowe PWN, 2007.

[82] J. P. W. Pluim, J. B. A. Maintz, M. A. Viergever. Mutual information based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 20, 2003.

[83] I. E. Pratikakis, H. Sahli, J. Cornelis. Low level image partitioning guided by the gradient watershed hierarchy. *Signal Processing*, 75(2):173–195, 1999.

[84] C. Rambabu, I. Chakrabarti. An efficient immersion-based watershed transform method and its prototype architecture. *Journal of Systems Architecture*, 53(4):210–226, 2007.

[85] M. E. Rettmann, X. Han, C. Xu, J. L. Prince. Automated sulcal segmentation using watersheds on the cortical surface. *NeuroImage*, 15(2):329–344, 2002.

[86] C. Revol, M. Jourlin. A new minimum variance region growing algorithm for image segmentation. *Pattern Recognition Letters*, 18:249–258, 1997.

[87] C. Revol-Muller, F. Peyrin, Y. Carrillon, C. Odet. Automated 3d region growing algorithm based on an assessment function. *Pattern Recognition Letters*, 23:137–150, 2002.

[88] J. F. Rivest, P. Soille, S. Beucher. Morphological gradients. *Electronic Imaging*, 2(4):326–336, 1993.

[89] R. Rodríguez, T. E. Alarcón, O. Pacheco. A new strategy to obtain robust markers for blood vessels segmentation by using the watersheds method. *Computers in Biology and Medicine*, 35(8):665–686, 2005.

[90] J. B. T. M. Roerdink, A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2001.

[91] H. C. Romesburg. *Cluster Analysis for Researchers*. Lulu Press, 2004.

[92] K. Saarinen. Color image segmentation by a watershed algorithm and region adjacency graph processing. *ICIP (3)*, strony 1021–1025, 1994.

[93] G. Sapiro, D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *Image Processing, IEEE Transactions on*, 5(11):1582–1586, 1996.

[94] R. Schettini. A segmentation algorithm for color images. *Pattern Recognition Letters*, 14:499–506, 1993.

[95] P. Scheunders, J. Sijbers. Multiscale watershed segmentation of multivalued images. *16th International Conference on Pattern Recognition*, wolumen 3, 2002.

[96] G. Sharma, H. J. Trussell. Digital color imaging. *IEEE Transactions on Image Processing*, 6(7):901–932, 1997.

[97] J. Sijbers, P. Scheunders, M. Verhoye, A. V. der Linden, D. van Dyck, E. Raman. Watershed-based segmentation of 3D MR data for volume quantization. *Magnetic Resonance Imaging*, 15(6):679–688, 1997.

[98] J. Smołka. Multi-class region growing algorithm. *Annales UMCS Informatica*, 1:165–173, 2003.

[99] J. Smołka. Multi-class 3d region growing algorithm. *Annales UMCS Informatica*, 2:227–236, 2004.

[100] J. Smołka. Szybki algorytm rozrostu regionu oparty na zlewiskach. *XIV Krajowa Konferencja Naukowa*, Biocybernetyka i inżynieria biomedyczna, 2005.

[101] J. Smołka. Watershed based region growing algorithm. *Annales UMCS Informatica*, 3:169–178, 2005.

[102] J. Smołka. Automatyczny zlewiskowy algorytm rozrostu regionu. *Algorytmy i programy*, Varia Informatica, strony 33–42, 2006.

[103] J. Smołka. Multilevel near optimal thresholding applied to watershed grouping. *Annales UMCS Informatica*, 5:191–200, 2006.

[104] J. Smołka. Hierarchical cluster analysis methods applied to image segmentation by watershed merging. *Annales UMCS Informatica*, AI 6:73–84, 2007.

[105] J. Smołka, S. Grzegórski. Optymalizacja algorytmu rozrostu regionu opartego na zlewiskach. *Obliczenia i technologie*, Varia Informatica, strony 23–32, 2005.

[106] J. Smołka, M. Skublewska-Paszkowska. Usuwanie nadmiernej segmentacji w transformacji wododziałowej za pomocą analizy skupień. *AGH Automatyka*, 12(2):463–482, 2008.

[107] J. Smołka, M. Skublewska-Paszkowska. Watershed merging method for color images. *Annales UMCS Informatica*, AI 8(1):111–121, 2008.

[108] J. Smołka, M. Skublewska-Paszkowska, A. Wojdyga. Improving performance of watershed clustering algorithm by using wavelet transform. *Polish Journal of Environmental Studies*, 18(3B):341–346, 2009.

[109] M. Sobczyk. *Statystyka - aspekty praktyczne i teoretyczne*. Wydawnictwo UMCS, 2006.

[110] P. Soille. Morphological carving. *Pattern Recognition Letters*, 25(5):543–550, 2004.

[111] M. Sonka, V. Hlavac, R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publising, wydanie drugie, 1998.

[112] C. Studholme, D. L. G. Hill, D. J. Hawkes. Incorporating connected region labelling into automatic image registration using mutual information. *MMBIA '96: Proceedings of the 1996 Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA '96)*, strona 23, Washington, DC, USA, 1996. IEEE Computer Society.

[113] H. Sun, J. Y. Yang, M. Ren. A fast watershed algorithm based on chain code and its application in image segmentation. *Pattern Recognition Letters*, 26(9):1266–1274, 2005.

[114] J.-P. Thiran, V. Warscotte, B. Macq. A queue-based region growing algorithm for accurate segmentation of multidimensional digital images. *Signal Processing*, 60(1):1–10, 1997.

[115] A. Trémeau, P. Colantoni. Regions adjacency graph applied to color image segmentation. *IEEE Trans. Image Processing*, 9(4):735–744, 2000.

[116] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2:176–201, 1993.

[117] L. Vincent, P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(6):583–598, 1991.

[118] N. Volkmann. A novel three-dimensional variant watershed transform for segmentation of electron density maps. *Journal of Structural Biology*, 138:123–129, 2002.

[119] D. Wang. A multiscale gradient algorithm for image segmentation using watersheds. *Pattern Recognition*, 30(12):2043–2052, 1997.

[120] S. K. Weeratunga, C. Kamath. A comparison of pde-based non-linear anisotropic diffusion techniques for image denoising. *Proceedings of SPIE Electronic Imaging, Image Processing: Algorithms and Systems II*, 2002.

[121] S. Wegner, T. Harms, H. Oswald, E. Fleck. Medical image segmentation using the watershed transformation on graphs. *ICIP*, strony III: 37–40, 1996.

[122] S. Wegner, T. Harms, H. Oswald, E. Fleck. The watershed transformation on graphs for the segmentation of CT images. *ICPR*, strony III: 498–502, 1996.

[123] R. T. Whitaker, X. Xue. Variable-conductance, level-set curvature for image denoising. *Proceedings of 3'rd International Conference on Image Processing*, strony 142–145, 2001.

[124] A. Wright, S. Acton. Watershed pyramids for edge detection. *ICIP*, strony II: 578–581, 1997.

[125] P.-Y. Yin, L.-H. Chen. A fast iterative scheme for multilevel thresholding methods. *Signal Processing*, 60(3):305–313, 1997.

[126] S. D. Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, 1986.

[127] H. Zhang, J. E. Fritts, S. A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.

[128] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.

[129] C. G. Zhao, T. G. Zhuang. A hybrid boundary detection algorithm based on watershed and snake. *Pattern Recognition Letters*, 26(9):1256–1265, 2005.

Dodatki

Dodatek A

Hierarchiczne metody analizy skupień

Metody analizy skupień służą do łączenia elementów w klastry (grupy). Obiekty należące do poszczególnych klastrów charakteryzują się stopniem podobieństwa, które jest określane za pomocą pewnej metryki (funkcji podobieństwa). Grupowanie obiektów odbywa się bez nadzoru. Wśród metod analizy skupień występują: metody hierarchiczne (aglomeracyjne i deglomeracyjne), metody K-średnich oraz metody rozmytej analizy skupień. W pracy wykorzystano aglomeracyjne metody hierarchiczne. Ta kategoria metod rozpoczyna od podziału, w którym każdy obiekt stanowi osobny klaster. W każdej kolejnej iteracji łączy dwa najbardziej podobne i kończy działanie, gdy wszystkie elementy znajdują się w tej samej grupie. W trakcie pracy algorytm realizujący analizę skupień tworzy hierarchię podobieństwa – stąd nazwa zastosowanej kategorii.

A.1 Etapy analizy skupień

Hierarchiczna analiza skupień składa się z wielu etapów. Niektóre z nich są niezbędne, a niektóre opcjonalne. To czy pewien etap jest wykonywany może zależeć od celu analizy, wymagań dotyczących zastosowanych współczynników oraz wybranej metody. Poniżej zostały opisane kolejne etapy hierarchicznej analizy skupień:

1. **Wybór zestawu atrybutów, które będą charakteryzowały każdy obiekt**. Atrybuty można podzielić na: ilościowe, jakościowe oraz porządkowe. W pracy stosowane są atrybuty ilościowe, ponieważ jest to najbardziej naturalny wybór w przypadku analizy obrazów cyfrowych. Wybrany zestaw

atrybutów jest dostosowany do rodzaju obrazu i obiektów, które przedstawia.

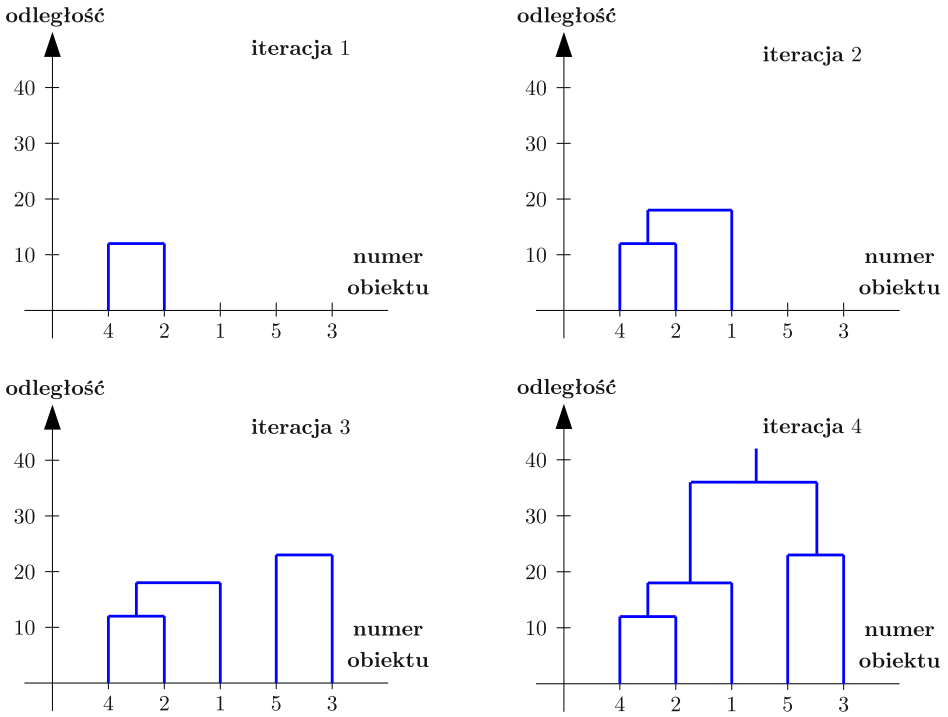
2. **Wyznaczenie wartości atrybutów dla każdego z obiektów**. Atrybuty, których wartości nie daje się obliczyć są oznaczane ponieważ liczba atrybutów opisujących obiekt jest istotna przy obliczaniu współczynników podobieństwa (punkt A.4). Wyniki tworzą macierz danych.

3. **Standaryzacja macierzy danych** jest krokiem opcjonalnym. Pozwala ona na ujednolicenie zakresów wartości poszczególnych atrybutów i zapobiega sytuacji, w której jeden atrybut ma dominujący wpływ na wartość współczynnika podobieństwa. Metody standaryzacji szczegółowo opisano w punkcie A.5.

4. **Obliczenie macierzy podobieństwa obiektów**. Jest to macierz kwadratowa zawierająca informację o stopniu podobieństwa wszystkich możliwych par obiektów. Ze względu na fakt, iż obiekty tworzą pary nieuporządkowane, obliczana jest tylko część nad lub pod główną przekątną. Podobieństwo określa się za pomocą współczynników szczegółowo opisanych w punkcie A.4. Ze względu na ograniczenia, które mogą je charakteryzować wybór konkretnego współczynnika może zależeć od danych wejściowych. Należy również zaznaczyć, że nie wszystkie metody wymagają obliczania macierzy podobieństwa np. metoda Warda.

5. Od tego momentu rozpoczyna się iteracyjna część analizy skupień – **znajdowane są dwa najbardziej podobne do siebie klastry** (początkowo każdy klaster zawiera tylko jeden obiekt) – bądź to na podstawie macierzy podobieństwa, bądź na podstawie innego kryterium charakterystycznego dla stosowanej metody (jak w przypadku metody Warda) .

6. **Połączenie znalezionych klastrów w nowy większy klaster**. Wiąże się to z koniecznością uaktualnienia hierarchii podobieństwa. Proces powstawania przykładowej hierarchii podobieństwa przedstawiono na rysunku A.1. Na dole rysunku umieszczone są numery obiektów. Linie poprowadzone od numerów łączą się na poziomie odpowiadającym wartości współczynnika podobieństwa, przy którym nastąpiło połączenie klastrów, tworząc w ten sposób drzewo. Oprócz modyfikacji hierarchii, konieczna jest również zmiana macierzy podobieństwa. Należy obliczyć podobieństwo nowego klastra do wszystkich pozostałych. Sposób obliczenia tego podobieństwa zależy od wybranej metody grupowania. Istnieją różne metody - między innymi metody SLINK, UPGMA oraz CLINK. Zostały one opisane w punkcie A.3. Jak wspomniano wcześniej w metodzie Warda nie używa się macierzy podobieństwa, ale hierarchia podobieństwa powstaje także w jej przypadku.



Rysunek A.1: Powstawanie przykładowej hierarchii podobieństwa przedstawionej jako drzewo (na podstawie [91]).

7. Jeżeli wszystkie obiekty nie znajdują się jeszcze w jednym klastrze, następuje **powrót do punktu 5**.

A.2 Ocena jakości analizy skupień

Jedną z możliwości oceny jakości analizy skupień jest zastosowanie współczynnika korelacji grupowania (ang. cophenetic correlation coefficient) [77]. Oblicza się go na podstawie macierzy podobieństwa/odległości i macierzy reprezentującej uzyskaną hierarchię podobieństwa. Kwadratową macierz podobieństwa/odległości $\mathbf{S}$ o rozmiarach $n \times n$ elementów przekształca się [32] w wektor postaci:

$$\mathbf{s} = [\mathbf{S}_{21}, \mathbf{S}_{31}, \ldots, \mathbf{S}_{n1}, \mathbf{S}_{32}, \mathbf{S}_{42}, \ldots, \mathbf{S}_{n2}, \ldots, \mathbf{S}_{n,n-1}]^T. \tag{A.1}$$

Wektor posiada tylko $m = n(n-1)/2$ wartości, ponieważ pary obiektów są nieuporządkowane i macierz podobieństwa/odległości jest symetryczna (często prezentuje się tylko część pod główną przekątną). Podobnie postępuje się z macierzą drzewa $\mathbf{D}$, która ma identyczne rozmiary jak macierz $\mathbf{S}$. Powstaje wektor:

$$\mathbf{d} = [\mathbf{D}_{21}, \mathbf{D}_{31}, \ldots, \mathbf{D}_{n1}, \mathbf{D}_{32}, \mathbf{D}_{42}, \ldots, \mathbf{D}_{n2}, \ldots, \mathbf{D}_{n,n-1}]^T. \tag{A.2}$$

Następnie oblicza się współczynnik korelacji według wzoru [109]:

$$r_{sd} = \frac{cov(\mathbf{s}, \mathbf{d})}{\sigma(\mathbf{s})\sigma(\mathbf{d})} = \frac{\sum_{i=1}^{m}(s_i - \bar{s})(d_i - \bar{d})}{\left[\sum_{i=1}^{m}(s_i - \bar{s})^2 \sum_{i=1}^{m}(d_i - \bar{d})^2\right]^{1/2}}, \tag{A.3}$$

gdzie: $\sigma(\mathbf{s})$ – odchylenie standardowe współrzędnych wektora $\mathbf{s}$, $\sigma(\mathbf{d})$ – odchylenie standardowe współrzędnych wektora $\mathbf{d}$, a $cov(\mathbf{s}, \mathbf{d})$ – kowariancja współrzędnych wektorów $\mathbf{s}$ i $\mathbf{d}$.

A.3 Metody grupowania

Istnieje wiele hierarchicznych metod grupowania. Do najpopularniejszych należą: *UPGMA* (ang. *unweighted pair-group method using arithmetic averages*) – metoda średniego wiązania, *SLINK* (ang. *single linkage*) – metoda pojedynczego wiązania, *CLINK* (ang. *complete linkage*) – metoda pełnego wiązania oraz *metoda Warda* – minimalnej wariancji. Trzy pierwsze metody są do siebie podobne. Schemat działania tych metod jest identyczny i został opisany w punkcie A.1. Różnice między nimi sprowadzają się do różnego sposobu obliczania odległości, jaka dzieli nowy klaster od pozostałych. Metoda Warda zostanie opisana osobno, ponieważ różni się od pozostałych w większym stopniu. W dalszej części rozdziału wykorzystano następujące oznaczenia: C_p, C_q – klastry p i q, m – numer obiektu należącego do klastra C_p, n – numer obiektu należącego do klastra C_q, d_{mn} – miara odległości między obiektami m i n, s_{mn} – miara podobieństwa obiektów m i n, $d_{C_pC_q}$ – miara odległości klastrów C_p i C_q, $s_{C_pC_q}$ – miara podobieństwa klastrów C_p i C_q, n_{C_p} – liczba obiektów w klastrze C_p, n_{C_q} – liczba obiektów w klastrze C_q,

Pierwsza metoda, czyli UPGMA, definiuje stopień podobieństwa (lub odległość) jako średnią arytmetyczną podobieństw (odległości) wszystkich możliwych par obiektów [91], które składają się z jednego obiektu wchodzącego w skład klastra o numerze p i drugiego wchodzącego w skład klastra q, przy czym $p \neq q$, co wyraża następujące równanie:

$$d_{C_pC_q} = \frac{1}{n_{C_p} \cdot n_{C_q}} \sum_{\substack{m \in C_p \\ n \in C_q}} d_{nm}; \quad p \neq q. \tag{A.4}$$

W przypadku wzoru A.4 nie ma znaczenia, czy wykorzystywany jest współczynnik podobieństwa, czy miara odległości, tak więc odległość d można zastąpić miarą podobieństwa s.

Metoda pojedynczego wiązania (SLINK) określa podobieństwo (odległość) dwóch klastrów p i q jako podobieństwo (odległość) dwóch najbardziej podobnych (najbliższych) obiektów [91], z których jeden należy do klastra p, a drugi do

klastra q. Tak jak w przypadku metody UPGMA $p \neq q$. Oznacza to, że wszystkie możliwe pary obiektów – pochodzących po jednym z łączonych klastrów – charakteryzują się nie większym stopniem podobieństwa. W przypadku stosowania współczynnika podobieństwa najbardziej podobne obiekty mają największą jego wartość:

$$s_{C_pC_q} = \max_{\substack{m \in C_p \\ n \in C_q}} s_{nm}; \quad p \neq q, \tag{A.5}$$

natomiast w przypadku miary odległości najbardziej podobne obiekty są najbliższymi sąsiadami:

$$d_{C_pC_q} = \min_{\substack{m \in C_p \\ n \in C_q}} d_{nm}; \quad p \neq q. \tag{A.6}$$

Ze sposobu w jaki metoda SLINK wyznacza podobieństwo klastrów (wybór najbliższych sąsiadów) wynika jej skłonność do budowania łańcuchów. Łańcuch powstaje, gdy przykładowo obiekty 1 i 2 łączone są w nowy klaster. W kolejnej iteracji do tego klastra dołączany jest obiekt 3, a w następnej iteracji 4 itd. Dołączane obiekty nie muszą mieć kolejnych numerów. Występowanie łańcuchów w hierarchii podobieństwa może uniemożliwiać odnalezienie wyraźnego podziału na klastry, może też oznaczać, że taki podział nie istnieje. Należy dodać, że inne metody również mogą tworzyć łańcuchy w hierarchii podobieństwa, jednak zjawisko to występuje rzadziej niż w przypadku metody SLINK [91].

Metoda pełnego wiązania (CLINK) jest przeciwieństwem metody SLINK. W tym wypadku podobieństwem (odległością) dwóch klastrów jest podobieństwo (odległość) dwóch najmniej podobnych obiektów [91] spośród wszystkich par obiektów należących do różnych klastrów. Oznacza to, że wszystkie pozostałe pary są podobne do siebie w większym stopniu. W przypadku stosowania współczynnika podobieństwa metoda szuka wartości minimalnej

$$s_{C_pC_q} = \min_{\substack{m \in C_p \\ n \in C_q}} s_{nm}; \quad p \neq q, \tag{A.7}$$

a w przypadku odległości maksymalnej

$$d_{C_pC_q} = \max_{\substack{m \in C_p \\ n \in C_q}} d_{nm}; \quad p \neq q. \tag{A.8}$$

Metoda Warda, w odróżnieniu od pozostałych, nie korzysta z macierzy podobieństwa, nie ma potrzeby jej uaktualniania. Nie korzysta ona również z zewnętrznych współczynników podobieństwa/różnicy, ponieważ sposób porównywania obiektów jest niejako "wbudowany" w metodę. W każdej iteracji metoda Warda łącząc kolejne obiekty w klastry, sprawdza wszystkie możliwe kombinacje łączenia klastrów i wybiera tę, która powoduje najmniejszy wzrost całkowitego błędu kwadratowego E w porównaniu z poprzednią iteracją [32]. Oznacza to, że w każdej iteracji metoda musi sprawdzić wszystkie możliwe sposoby połączenia

istniejących klastrów. Należy również zauważyć, że metoda Warda nie gwarantuje uzyskania optymalnego zgrupowania obiektów w klastry tzn. takiego, który będzie charakteryzował się minimalnym błędem E. Wynika to z faktu, iż raz połączone klastry nigdy nie są dzielone. Całkowity błąd E dany jest wzorem:

$$E = \sum_{p=1}^{g} E_p, \tag{A.9}$$

gdzie błąd klastra o numerze p wynosi:

$$E_p = \sum_{k=1}^{n_p} \sum_{i=1}^{n} (x_{i,kp} - \bar{x}_{i,p})^2 \tag{A.10}$$

W powyższych wzorach przyjęto następujące oznaczenia: g – liczba klastrów, n – liczba atrybutów charakteryzujących obiekty, n_p – liczba obiektów w p-tym klastrze, $\bar{x}_{i,p}$ – wartość średnia i-tego atrybutu w p-tym klastrze, $x_{i,kp}$ – wartość i-tego atrybutu k-tego obiektu w p-tym klastrze.

A.4 Współczynniki podobieństwa i miary odległości

Wszystkie opisane powyżej metody, z wyjątkiem metody Warda, wymagają obliczenia macierzy podobieństwa/odległości. Do jej utworzenia konieczne są odpowiednie współczynniki pozwalające na określenie stopnia podobieństwa obiektów. Metody CLINK, UPGMA oraz SLINK pozwalają na elastyczny wybór takiego współczynnika w zależności od rodzaju danych wejściowych czy pożądanego efektu. Odpowiedni wybór współczynnika pozwala na przykład na ignorowanie różnic określonego rodzaju między obiektami. W dalszej części rozdziału zastosowano następujące oznaczenia: j, k – numery obiektów, n – liczba atrybutów, x_{ij} – wartość i-tego atrybutu j-tego obiektu, x_{ik} – wartość i-tego atrybutu k-tego obiektu

Pierwszym i najprostszym współczynnikiem jest *odległość euklidesowa* e_{jk}. Jest to oczywiście miara odległości [91], co oznacza, że mniejsza wartość to większy stopień podobieństwa obiektów. Gdy odległość wynosi 0 obiekty są identyczne. Odległość dwóch punktów w n-wymiarowej przestrzeni wyrażona jest wzorem:

$$e_{jk} = \left[\sum_{i=1}^{n} (x_{ij} - x_{ik})^2 \right]^{1/2} \tag{A.11}$$

i przyjmuje wartości $0 \leqslant e_{jk} < \infty$.

Kolejnym współczynnikiem jest *średnia odległość euklidesowa* d_{jk}. Jest to również miara odległości [91] przyjmująca wartości z zakresu $0 \leqslant d_{jk} < \infty$. Wyrażona jest wzorem:

$$d_{jk} = \left[\sum_{i=1}^{n} (x_{ij} - x_{ik})^2 / n \right]^{1/2} \tag{A.12}$$

Jak widać $d_{jk} = e_{jk}/\sqrt{n}$ i jeżeli dla wszystkich par obiektów wartość n jest identyczna, to uzyskana hierarchia podobieństwa będzie różniła się od hierarchii uzyskanej za pomocą odległości euklidesowej tym, że poziomy, przy których następuje łączenie klastrów, będą miały wartości podzielone przez $\sqrt{n}$ [91]. Użycie współczynnika średniej odległości euklidesowej jest uzasadnione w wypadku, gdy nie wszystkie obiekty posiadają taką samą liczbę atrybutów. Taka sytuacja może mieć miejsce, gdy dla określonego obiektu nie można wyznaczyć wszystkich atrybutów.

Tak jak poprzednie *współczynnik różnicy kształtu* jest miarą odległości [91] przyjmującą wartości $0 \leqslant z_{jk} < \infty$. Współczynnik ten określa równanie:

$$z_{jk} = \left[\frac{n}{n-1}\left(d_{jk}^2 - q_{jk}^2\right)\right]^{1/2}, \tag{A.13}$$

gdzie:

$$q_{jk}^2 = \frac{1}{n^2}\left(\sum_{i=1}^{n} x_{ij} - \sum_{i=1}^{n} x_{ik}\right). \tag{A.14}$$

Współczynnik różnicy kształtu może przyjmować wartość 0, nawet gdy wszystkie odpowiadające atrybuty dwóch różnych obiektów różnią się o tę samą stałą (innymi słowy ignoruje translację wartości atrybutów). W przeciwnym wypadku przyjmuje wartości dodatnie.

Współczynnik kosinusowy jest współczynnikiem podobieństwa [91] przyjmującym wartości $-1 \leqslant c_{jk} \leqslant 1$. Wartość maksymalna 1 oznacza największe podobieństwo, podczas gdy -1 oznacza jego brak. Wzór:

$$c_{jk} = \frac{\sum_{i=1}^{n} x_{ij}x_{ik}}{\left(\sum_{i=1}^{n} x_{ij}^2\right)^{1/2}\left(\sum_{i=1}^{n} x_{ik}^2\right)^{1/2}} \tag{A.15}$$

jest wyrażeniem na kosinus kąta między dwoma wektorami w n-wymiarowej przestrzeni. Ponieważ długości wektorów nie mają wpływu na kosinus kąta między nimi, współczynnik kosinusowy ignoruje mnożenie wartości wszystkich atrybutów przez stałą (ignoruje skalowanie). Czasami wygodniej posługiwać się miarą odległości niż współczynnikiem podobieństwa. Współczynnik kosinusowy łatwo zmodyfikować, tak aby większe wartości oznaczały mniejszy stopień podobieństwa [32]. Przyjęcie zależności:

$$c'_{jk} = \frac{1 - c_{jk}}{2} \tag{A.16}$$

powoduje, że współczynnik przyjmuje wartości $0 \leqslant c'_{jk} \leqslant 1$, przy czym 0 oznacza największe podobieństwo a 1 najmniejsze.

Współczynnik korelacji liniowej Pearsona [81] przyjmuje, podobnie jak współczynnik kosinusowy, wartości $-1 \leqslant r_{jk} \leqslant 1$ i też jest współczynnikiem podobieństwa, co oznacza, że wartość maksymalna, czyli 1, oznacza największy stopień podobieństwa. Współczynnik korelacji dany jest wzorem:

$$r_{jk} = \frac{\sum_{i=1}^{n} x_{ij}x_{ik} - \frac{1}{n}\left(\sum_{i=1}^{n} x_{ij}\right)\left(\sum_{i=1}^{n} x_{ik}\right)}{\left\{\left[\sum_{i=1}^{n} x_{ij}^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_{ij}\right)^2\right]\left[\sum_{i=1}^{n} x_{ik}^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_{ik}\right)^2\right]\right\}^{1/2}}. \tag{A.17}$$

Należy zauważyć, iż wartość 1 nie oznacza, że porównywane obiekty mają identyczne atrybuty. Ponadto współczynnik korelacji jest, podobnie jak współczynnik różnicy kształtu, całkowicie niewrażliwy na translacje. Podobnie jak współczynnik kosinusowy również współczynnik korelacji można zmodyfikować za pomocą równania:

$$r'_{jk} = \frac{1 - r_{jk}}{2}, \tag{A.18}$$

tak aby stał się miarą odległości [32] i zawierał się w przedziale: $0 \leqslant r'_{jk} \leqslant 1$.

Następny współczynnik czyli *metryka Canberry* jest miarą odległości [91] przyjmującą wartości z zakresu $0 \leqslant a_{jk} \leqslant 1$. Przy czym 0 oznacza, że wartości atrybutów są identyczne. Współczynnik wyraża się wzorem:

$$a_{jk} = \frac{1}{n}\sum_{i=1}^{n} \frac{|x_{ij} - x_{ik}|}{x_{ij} + x_{ik}} \tag{A.19}$$

Charakterystyczną cechą metryki Canberry jest to, że wszystkie składniki sumy, odzwierciedlające różnice między atrybutami, skalowane są do przedziału od 0 do 1 [91]. Dzięki temu atrybuty znacząco większe od pozostałych nie mają dominującego wpływu na ostateczną wartość. Czynnik $1/n$ przed sumą pozwala skompensować różnice w liczbie atrybutów, które można było określić. Współczynnik ten wymaga, aby wartości atrybutów były nieujemne.

Współczynnik Bray'a-Curtisa jest również miarą odległości [91]. Przyjmuje wartości z zakresu $0 \leqslant b_{jk} \leqslant 1$. Wartość 0 oznacza sytuację, w której atrybuty obiektów j i k są identyczne. Wzór:

$$b_{jk} = \frac{\sum_{i=1}^{n} |x_{ij} - x_{ik}|}{\sum_{i=1}^{n} (x_{ij} + x_{ik})} \tag{A.20}$$

jest dość podobny do wzoru na metrykę Canberry, jednak w tym wypadku wpływ poszczególnych atrybutów na ostateczną wartość współczynnika nie jest wyrównywany. Podobnie jak w przypadku metryki Canberry, wartości atrybutów powinny być nieujemne.

Ostatnim omówionym współczynnikiem jest błąd kwadratowy "wbudowany" w metodę Warda. Błąd definiuje równanie A.9. Jest to współczynnik różnicy wrażliwy na translacje oraz skalowanie wartości atrybutów [91].

A.5 Metody standaryzacji

Atrybuty obiektów mogą przyjmować wartości należące do różnych zakresów. Wartości te mogą się różnić nawet o kilka rzędów wielkości. Może to powodować, że wartość współczynnika podobieństwa będzie zależała, w praktyce, tylko od jednego atrybutu, który przyjmuje wartości znacząco większe od pozostałych. Spośród współczynników podobieństwa wymienionych w punkcie A.4 tylko metryka Canberry zapewnia wyrównywanie wpływu poszczególnych atrybutów na ostateczną wartość współczynnika podobieństwa/różnicy. W przypadku stosowania pozostałych atrybutów, jeżeli opisana powyżej sytuacja jest niepożądana, konieczne jest użycie jednej z metod standaryzacji atrybutów. Poniżej opisano cztery różne metody. We wzorach użyto następujących oznaczeń: j, k – numery obiektów, t – liczba obiektów, x_{ij} – wartość i-tego atrybutu j-tego obiektu, x_{ik} – wartość i-tego atrybutu k-tego obiektu.

Pierwsza z metod standaryzacji do obliczania nowych wartości atrybutów z_{ij} wykorzystuje równanie [109]:

$$z_{ij} = \frac{x_{ij} - \bar{x}_i}{\sigma_i}, \tag{A.21}$$

gdzie średnia $\bar{x}_i$ i-tego atrybutu wynosi:

$$\bar{x}_i = \frac{\sum_{j=1}^{t} x_{ij}}{t}, \tag{A.22}$$

a jego odchylenie standardowe:

$$\sigma_i = \left[\left(\sum_{j=1}^{t} (x_{ij} - \bar{x}_i)^2 \right) / (t-1) \right]^{1/2}. \tag{A.23}$$

Nowe wartości atrybutów obliczone za pomocą równania A.21 będą zawsze miały średnią $\bar{z}_i = 0$ i odchylenie standardowe $\sigma_i' = 1$. Jeżeli wartości oryginalnych atrybutów są niezerowe, to na pewno część wartości atrybutów z_{ij} będzie przyjmować wartości ujemne. Oznacza to, że tej metody standaryzacji nie można wykorzystywać w połączeniu ze współczynnikami podobieństwa/różnicy, które nie dopuszczają ujemnych wartości atrybutów (np. metryka Canberry, współczynnik Bray'a-Curtisa)

Drugim sposobem ujednolicenia atrybutów jest zastosowanie równania [91]:

$$z_{ij} = \frac{x_{ij}}{x_{max\,i}}, \tag{A.24}$$

które powoduje, że ustandaryzowane wartości będą spełniały zależność $0 \leqslant z_{ij} \leqslant 1$ pod warunkiem, że wartości oryginalnych atrybutów są większe lub równe 0. Wartości z_{ij} odpowiadające maksymalnym wartościom x_{ij} będą równe 1. Wartości odpowiadające minimom nie muszą być równe 0.

Kolejny sposób standaryzcji jest podobny. Użycie równania [91]:

$$z_{ij} = \frac{x_{ij} - x_{min\,i}}{x_{max\,i} - x_{min\,i}} \tag{A.25}$$

spowoduje, że wartości każdego z atrybutów będą należały do przedziału $0 \leqslant z_{ij} \leqslant 1$, przy czym w odróżnieniu od równania A.24 wartości minimalne oryginalnych atrybutów odpowiadają wartościom $z_{ij} = 0$.

Ostatnia z opisanych metod, wyrażona wzorem [91]:

$$z_{ij} = \frac{x_{ij}}{\sum_{i=1}^{t} x_{ij}} \tag{A.26}$$

powoduje, że wartości z_{ij} będą spełniały zależność $\sum_{i=1}^{t} z_{ij} = 1$.

Spis skrótów

Skrót	Opis
CIELab	przestrzeń barw opracowana przez Commission Internationale d'Eclairage posiadająca składowe: L*, a* i b*
HSV	przestrzeń barw o trzech składowych: barwie, nasyceniu i wartości (ang. hue, saturation, and value color space)
RGB	przestrzeń barw o trzech składowych: czerwonej, zielonej i niebieskiej (ang. red, green, blue color space)
sRGB	ustandaryzowane RGB (ang. standard RGB)
CLINK	metoda pełnego wiązania (ang. complete linkage)
SLINK	metoda pojedynczego wiązania (ang. single linkage)
UPGMA	metoda średniego wiązania (ang. unweighted pair-group method using arithmetic averages)
CT	tomografia komputerowa (ang. computed tomography)
MR	rezonans magnetyczny (ang. magnetic resonance)
MRI	obrazowanie rezonansu magnetycznego (ang. magnetic resonance imaging)
MRI PD	obraz MRI przedstawiający gęstość protonów (ang. proton density weighted MRI)
MRI T1	obraz MRI T1-zależny (ang. T1-weighted MRI)
MRI T2	obraz MRI T2-zależny (ang. T2-weighted MRI)
DTF	dynamiczny filtr sekwencji w czasie (ang. dynamic time filter)
filtr top-hat	filtr cylindra
gradient-pca	gradient obliczany jako różnica największych wartości własnych macierzy w analizie głównych składowych
gradient-skp	gradient zdefiniowany jako pierwiastek sumy kwadratów pochodnych cząstkowych
TC	operacja domknięcia w czasie (ang. time closing)
TO	operacja otwarcia w czasie (ang. time opening)

Skrót	Opis
FIFO	kolejka – bufor typu pierwszy na wejściu, piewszy na wyjściu (ang. first in, first out)
LUT	tabela podglądu (ang. look up table)
MSE	błąd średniokwadratowy (ang. mean square error)
VTK	biblioteka wizualizacyjna (ang. visualization toolkit)

Spis rysunków

2.1 Typowy sposób obliczania transformacji wododziałowej 5
2.2 Główne kategorie algorytmów realizujących transformację wododziałową 10
2.3 Maski wykorzystywane przy wyznaczaniu obrazu gradientowego 22
2.4 Obraz gradientowy obliczony dla obrazu monochromatycznego 22
2.5 Ograniczenia filtru gradientowego wysokiej rozdzielczości 23
2.6 Obrazy gradientowe obliczony dla obrazu barwnego 24
2.7 Transformacja wododziałowa bez redukcji nadsegmentacji 27
2.8 Przegląd metod redukcji nadsegmentacji 28
2.9 Transformacja wododziałowa na grafie 37
2.10 Algorytm wodospadów 39
2.11 Rozrost regionu na grafie 41

3.1 Przygotowanie obrazu dla analizy skupień 50
3.2 Wpływ zastosowania selektywnego filtru rozmywającego na nadsegmentację 52
3.3 Zmniejszanie nadsegmentacji poprzez progowanie 53
3.4 Wpływ progowania na nadsegmentację 54
3.5 Zmniejszanie nadsegmentacji poprzez zalewanie płytkich zlewisk . . . 55
3.6 Wpływ poziomu łączenia płytkich zlewisk na nadsegmentację 60
3.7 Atrybuty zlewisk (1) 62
3.8 Atrybuty zlewisk (2) 63
3.9 Obliczenie hierarchii podobieństwa zlewisk 66
3.10 Uzyskanie ostatecznej segmentacji 71
3.11 Podział hierarchii podobieństwa 72
3.12 Wyniki redukcji nadsegmentacji 74

4.1 Podział metod oceny segmentacji - Y.J.Zhang 77
4.2 Podział metod oceny segmentacji - H.Zhang 78
4.3 Monochromatyczne obrazy testowe 87
4.4 Wybrane segmentacje obrazów CT 87

4.5 Wybrane segmentacje obrazów MRI 88
4.6 Histogramy obrazów barwnych 91
4.7 Wybrane segmentacje obrazów barwnych (1) 99
4.8 Wybrane segmentacje obrazów barwnych (2) 100

5.1 Przyspieszanie segmentacji za pomocą transformaty falkowej 103
5.2 Kolejne kroki transformaty falkowej 105
5.3 Wpływ skali na nadsegmentację 106
5.4 Przywracanie oryginalnej rozdzielczości 107
5.5 porównanie filtrów krawędziowych 109
5.6 Wyniki segmentacji w różnych skalach 114

6.1 Różne sposoby przedstawiania obrazu nadmiernie posegmentowanego . 118
6.2 Przykładowa segmentacja obrazu w przestrzeni Lab 119
6.3 Przykładowa segmentacja zaszumionych obszarów w obrazie 120
6.4 Przykładowa segmentacja bez uwzględniania jasności barwy 120
6.5 Przykładowa segmentacja uwzględniająca różnice jasności 121
6.6 Wybór liczby klas po utworzeniu hierarchii podobieństwa 122
6.7 Wykres funkcji oceniajacych F i Q dla obrazu testowego 123
6.8 Przykładowe segmentacje wybrane na podstawie funkcji F i Q (1) . . 123
6.9 Przykładowe segmentacje wybrane na podstawie funkcji F i Q (2) . . 124
6.10 Wybór liczby kroków transformaty falkowej 125
6.11 Porównanie segmentacji z obrazem oryginalnym 125
6.12 Wybór wielu poziomów przecięcia hierarchii podobieństwa 126
6.13 Podział klasy na mniejsze części 127
6.14 Łączenie klas 127

A.1 Powstawanie hierarchii podobieństwa 144

Spis tabel

2.1 Oznaczenia w algorytmie analizy skupień 14

3.1 Dodatkowe oznaczenia w algorytmie wstępnej redukcji nadsegmentacji 55
3.2 Dodatkowe oznaczenia w algorytmie generującym hierarchię podobieństwa zlewisk 66
3.3 Dodatkowe oznaczenia w algorytmie generującym tablice LUT 72

4.1 Zestawienie najlepszych parametrów analizy skupień - skala szarości . 84
4.2 Zestawienie najlepszych parametrów analizy skupień - obrazy barwne 96

5.1 Porównanie czasów segmentacji wykorzystującej transformatę falkową 111