

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I ELEKTRONIKI

Katedra Automatyki

ROZPRAWA DOKTORSKA

**Algorytmy komputerowej detekcji dużych obiektów  
w obrazach o wysokim poziomie szumu  
i niejednorodności**

mgr inż. Rafał Petryniak

Promotor  
Prof. dr hab. inż. Leszek Wojnar  
*Politechnika Krakowska  
im. Tadeusza Kościuszki*

Kraków 2011

*Pragnę serdecznie podziękować mojemu promotorowi  
prof. dr hab. inż. **Leszkowi Wojnarowi**,  
za życzliwą opiekę  
i pomoc w przygotowaniu niniejszej rozprawy doktorskiej.*

*Dziękuję wszystkim,  
którzy przyczynili się do powstania tej pracy.*

*Rafał Petryniak*

*Karolowi i Dorocie,  
którzy przyszli na świat  
w trakcie pisania tej pracy*

---

## *Spis treści*

<b>Spis treści</b>	<b>iii</b>
<b>Spis rysunków</b>	<b>v</b>
<b>Spis tabel</b>	<b>viii</b>
<b>Wykaz skrótów i oznaczeń</b>	<b>ix</b>
<b>I. Wprowadzenie</b>	
1. Wstęp	2
2. Analiza problemu, cel i zakres pracy	12
<b>II. Przygotowanie obrazu do dalszej analizy</b>	
3. Metody usuwania szumu	22
3.1. Proste filtry korzystające z sąsiedztwa punktu . . . . .	23
3.2. Liniowe wygładzenie obrazu bazujące na funkcji Gaussa . . . . .	24
3.3. Nieliniowe wygładzenie obrazu bazujące na funkcji Gaussa . . . . .	27
4. Wstępna segmentacja obrazu	33
4.1. Klasyczny algorytm działów wodnych . . . . .	34
4.2. Segmentacja z wykorzystaniem minimalnego drzewa rozpinającego . . . . .	36
5. Łączenie segmentów	39
5.1. Wykorzystanie reprezentacji grafowej na potrzeby algorytmu łączenia segmentów . . . . .	39
5.2. Popularne kryteria łączenia segmentów . . . . .	41
<b>III. Algorytmy segmentacji oraz detekcji dużych obiektów</b>	
6. Segmentacja obrazu z użyciem modelu bazującego na krawędzi	48
6.1. Problem nieciągłości krawędzi i sposoby ich łączenia . . . . .	48

6.2. Propozycja algorytmu łączenia regionów z wykorzystaniem informacji o krawędzi . . . . .	56
<b>7. Detekcja obiektów z użyciem modelu bazującego na kształcie</b>	<b>68</b>
7.1. Segmentacja obiektów na obrazie z wykorzystaniem atlasu . . . . .	69
7.2. Propozycja algorytmu dokładnego dopasowania modelu do obiektu . . . . .	71
<b>8. Interaktywna metoda segmentacji obrazu bazująca na modelu znaczników kontrolnych</b>	<b>93</b>
8.1. Algorytm działów wodnych bazujący na znacznikach . . . . .	94
8.2. Hierarchiczna segmentacja obrazu . . . . .	97
8.3. Podsumowanie . . . . .	99
 <b>IV. Podsumowanie</b>	
<b>9. Propozycje rozszerzające możliwości proponowanej metody</b>	<b>102</b>
9.1. Projekt automatycznego systemu rozpoznawania i detekcji dużych obiektów	103
9.2. Koncepcja systemu . . . . .	103
9.3. Realizacja systemu . . . . .	107
9.4. Podsumowanie prac nad projektem . . . . .	109
<b>10. Podsumowanie i wnioski</b>	<b>110</b>
 <b>V. Dodatki</b>	
<b>A. Algorytmy detekcji krawędzi</b>	<b>115</b>
A.1. Proste sposoby detekcji krawędzi . . . . .	115
A.2. Zaawansowana detekcja krawędzi na przykładzie <i>algorytmu Canny</i> . . . . .	117
<b>B. Algorytmy wyznaczające transformację jednego obrazu w drugi</b>	<b>120</b>
B.1. Techniki śledzenia obiektów . . . . .	121
B.2. Algorytmy nakładania obrazów . . . . .	124
<b>Literatura</b>	<b>129</b>
<b>Źródła internetowe</b>	<b>137</b>
<b>Skorowidz</b>	<b>138</b>

---

## *Spis rysunków*

1.1	Spektrum elektromagnetyczne z zaznaczonym zakresem promieniowania widzialnego . . . . .	3
1.2	Zdjęcia przykładowych miast ilustrujące problemy detekcji dużych obiektów . . . . .	7
1.3	Obrazy prezentujące przykłady dużych obiektów . . . . .	9
2.1	Wskazanie problemów w analizie dużych obiektów na przykładzie obrazu kości udowej. . . . .	13
2.2	Segmentacja radiogramu kości udowej za pomocą standardowych metod przetwarzania obrazu . . . . .	15
2.3	Możliwości redukcji liczby segmentów poprzez zastosowanie algorytmu łączącego segmenty . . . . .	17
3.1	Wyniki działania liniowych filtrów wygładzających . . . . .	25
3.2	Wykres dwuwymiarowej funkcji rozkładu Gaussa . . . . .	26
3.3	Wykres zależności współczynników wygładzenia obrazu od wartości estymatora krawędzi . . . . .	28
3.4	Wyniki działania nieliniowego filtra rozmycia obrazu opartego o <i>model Perona-Malik</i> . . . . .	29
3.5	Wygładzenie obrazu filtrem medianowym (rezultaty i profil) . . . . .	31
3.6	Porównanie profilu filtracji średnią arytmetyczną z medianą . . . . .	32
3.7	Porównanie profilu filtracji medianowej i filtracji <i>Perona-Malik</i> z obrazem oryginalnym . . . . .	32
4.1	Wyniki segmentacji kości biodrowej z wykorzystaniem algorytmu działów wodnych . . . . .	35
4.2	Segmentacja wododziałowa obrazu po uprzednim usunięciu szumu . . . . .	35
4.3	Przykład segmentacji obrazu niejednorodnego algorytmem grafowym poszukującym minimalnego drzewa rozpinającego . . . . .	37
4.4	Segmentacja rentgenowskich obrazów kości udowej algorytmem grafowym <i>MST</i> . . . . .	38
5.1	Testowa aplikacja prezentująca działanie różnych kryteriów łączenia segmentów . . . . .	43
5.2	Wyniki łączenia segmentów dla wybranego obrazu klatki piersiowej . . . . .	44
5.3	Wyniki łączenia segmentów dla wybranego obrazu kości udowej . . . . .	45

6.1	Problem segmentacji obrazu z użyciem obrazu krawędziowego . . . . .	49
6.2	Problem segmentacji obrazu z wykorzystaniem obrazu krawędziowego . . . . .	49
6.3	Uzupełnienie linii krawędzi metodą śledzenia z wykorzystaniem obrazu gradientowego . . . . .	51
6.4	Mapa odległości, jej negatyw i lokalne minima dla obrazu krawędziowego . . . . .	52
6.5	Różne warianty obrazów będących podstawą zalewania w algorytmie działów wodnych . . . . .	53
6.6	Wyniki łączenia krawędzi z wykorzystaniem algorytmu działów wodnych . . . . .	53
6.7	Prezentacja wykorzystania działów wodnych do zamykania krawędzi . . . . .	54
6.8	Przykładowy obraz echokardiograficzny serca oraz wyniki detekcji lewej komory . . . . .	55
6.9	Ilustracja wpływu obrazu gradientowego na wynik detekcji działów wodnych . . . . .	55
6.10	Obraz oryginalny i obrazy wejściowe do algorytmu łączenia regionów na podstawie krawędzi . . . . .	57
6.11	Pojęcie granicy oraz krawędzi między dwoma segmentami . . . . .	58
6.12	Klasyfikacja granic segmentów ze względu na sąsiedztwo krawędzi . . . . .	58
6.13	Wynik końcowy i wyniki pośrednie proponowanego algorytmu łączenia segmentów . . . . .	59
6.14	Przykład niespójności wyników algorytmu działów wodnych i detekcji krawędzi <i>Canny</i> . . . . .	60
6.15	Przykładowy schemat podziału segmentów wyrażony w postaci grafu (1) . . . . .	61
6.16	Przykładowy schemat podziału segmentów wyrażony w postaci grafu (2) . . . . .	64
6.17	Przykłady działania algorytmu łączenia segmentów na podstawie krawędzi dla innych zdjęć . . . . .	65
7.1	Przykładowe zdjęcia tomograficzne klatki piersiowej . . . . .	69
7.2	Pierwszy etap segmentacji bazującej na atlasie . . . . .	70
7.3	Drugi etap segmentacji bazującej na atlasie . . . . .	70
7.4	Trzeci etap segmentacji bazującej na atlasie . . . . .	71
7.5	Problemy z dokładnością algorytmu nakładania obrazów . . . . .	72
7.6	Obrazy wejściowe do proponowanego algorytmu dokładnego dopasowania . . . . .	73
7.7	Wyniki działania proponowanego algorytmu dokładnego dopasowania dla różnych wartości parametru <i>sigma</i> rozmycia Gaussa . . . . .	77
7.8	Wynik działania proponowanego algorytmu dokładnego dopasowania dla zdjęć tomograficznych klatki piersiowej . . . . .	79
7.9	Wzorcowy obraz kości udowej i manualnie przygotowany jej obrys . . . . .	80
7.10	Prezentacja działania proponowanego algorytmu dokładnego dopasowania dla obrazów kości udowej . . . . .	81
7.11	Poprawa detekcji kości udowej z zastosowaniem algorytmu łączenia segmentów . . . . .	83

7.12	Detekcja kości łokciowej z wykorzystaniem metod zaproponowanych w pracy . . . . .	85
7.13	Detekcja jamy miednicy z wykorzystaniem metod zaproponowanych w pracy . . . . .	86
7.14	Detekcja drugiej kości śródreżca lewej ręki z wykorzystaniem metod zaproponowanych w pracy . . . . .	87
7.15	Detekcja kości piszczelowej z wykorzystaniem metod zaproponowanych w pracy . . . . .	88
7.16	Detekcja kości udowej z wykorzystaniem metod zaproponowanych w pracy . . . . .	89
7.17	Detekcja kości piętowej z wykorzystaniem metod zaproponowanych w pracy . . . . .	90
7.18	Detekcja kości ramiennej z wykorzystaniem metod zaproponowanych w pracy . . . . .	91
8.1	Interaktywna segmentacja obrazu algorytmem działów wodnych z zastosowaniem znaczników punktowych . . . . .	95
8.2	Wpływ położenia punktu startowego na wynik segmentacji obrazu algorytmem działów wodnych . . . . .	96
8.3	Możliwości dokładnego określania obiektów do segmentacji z zastosowaniem znaczników ciągłych . . . . .	97
8.4	Proces hierarchicznej segmentacji obrazu z wykorzystaniem algorytmu działów wodnych na sztucznym obrazie . . . . .	98
8.5	Przykłady działania algorytmu działów wodnych ze znacznikiem na obrazach przedstawiających naturę . . . . .	100
9.1	Projekt graficznej bazy danych - pierwsze dodanie obrazów do bazy . . . . .	105
9.2	Projekt graficznej bazy danych - dodanie kolejnych obrazów do bazy . . . . .	106
9.3	Przykładowy zrzut ekranu przedstawiający działanie skonstruowanej graficznej bazy danych . . . . .	108
A.1	Detekcja krawędzi z użyciem obrazu gradientowego . . . . .	116
A.2	Redukcja szumu filtrem Gaussa o wartości $\sigma = 2.0$ . . . . .	118
A.3	Kierunek i zwrot gradientu . . . . .	118
A.4	Wynik eliminacji pikseli o niemaksymalnej wartości . . . . .	119
A.5	Obraz wynikowy dla algorytmu detekcji krawędzi <i>algorytmem Canny</i> . . . . .	119
B.1	Prezentacja działania algorytmu przepływu optycznego <i>Lukas-Kanade</i> . . . . .	123
B.2	Prezentacja działania algorytmu przepływu optycznego <i>Horn-Schunck</i> . . . . .	123
B.3	Prezentacja działania korelacyjnego <i>algorytmu dopasowania bloków</i> . . . . .	123
B.4	Prezentacja działania algorytmu nakładania obrazów bazującego na ruchu poziomic . . . . .	126
B.5	Prezentacja działania dwukierunkowego algorytmu nakładania obrazów z wykorzystaniem krzywych B-Spline . . . . .	127
B.6	Rezultaty nakładania obrazu uzyskane narzędziem <i>bUnwarpJ</i> . . . . .	128



---

*Spis tabel*

3.1	Macierz filtru uśredniającego o rozmiarach 7 x 7 . . . . .	24
3.2	Macierz filtru Gaussa o wartości $\sigma = 1$ . . . . .	27
A.1	Macierze filtrów gradientowych w kierunku poziomym i pionowym . . . . .	115

---

## Wykaz skrótów i oznaczeń

- $IM_{in}$  – analizowany obraz wejściowy
- $IM_{out}$  – obraz wynikowy
- $IM_{blur}$  – obraz po wygładzeniu
- $IM_{ws}$  – obraz reprezentujący segmenty uzyskane algorytmem działów wodnych
- $G(x, y)$  – dwuwymiarowa funkcja rozkładu Gaussa
- $\sigma$  – odchylenie standardowe rozkładu normalnego (*sigma*)
- $E(x, y)$  – estymator krawędzi
- $\nabla u(x, y)$  – gradient jasności obrazu
- $C_i$  – zbiór pikseli tworzących jeden segment na obrazie
- $Int(C_i)$  – wewnętrzne zróżnicowanie podzbioru  $C_i$
- $MST(C_i)$  – minimalne drzewo rozpinające podzbioru  $C_i$
- $e$  – oznaczenie krawędzi grafu
- $w(e)$  – waga krawędzi grafu  $e$
- $S, S_1, S_2, \dots$  – oznaczenie bieżącego i kolejnych analizowanych segmentów
- $K_i(S_1, S_2)$  – kryterium łączenia (podobieństwa) segmentów  $S_1$  i  $S_2$
- $M_i(S)$  – miara spójności segmentu  $S$
- $Im_{in}[x], x \in S$  – kolejne piksele segmentu  $S$
- $K_{ab}$  – kryterium łączenia segmentu  $a$  z segmentem  $b$
- $K_{ba}$  – kryterium łączenia segmentu  $b$  z segmentem  $a$
- $dg_{ab}$  – długość granicy pomiędzy segmentami  $a$  i  $b$
- $dg_{ba}$  – długość granicy pomiędzy segmentami  $b$  i  $a$
- $dk_{ab}$  – długość krawędzi na granicy segmentów  $a$  i  $b$
- $dk_{ba}$  – długość krawędzi na granicy segmentów  $b$  i  $a$
- $Op_a$  – obwód segmentu  $a$
- $Op_b$  – obwód segmentu  $b$

# Część I

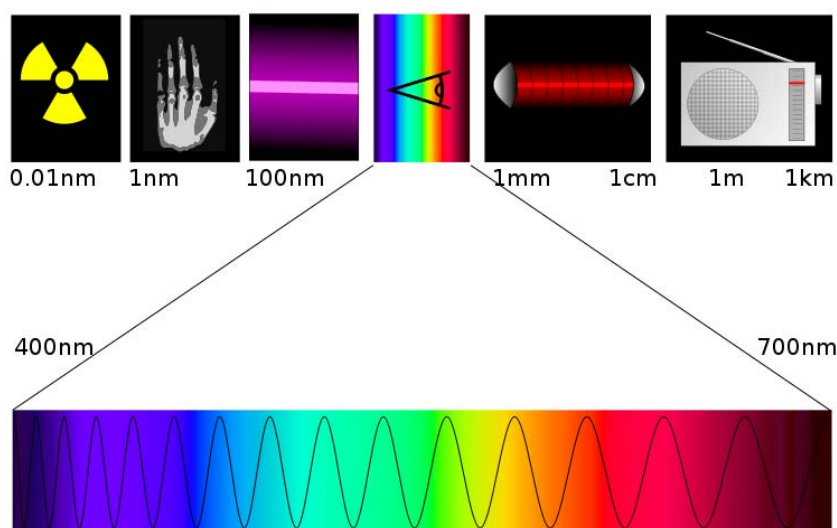
## Wprowadzenie

## Rozdział 1

---

### *Wstęp*

Wraz z pojawieniem się pierwszych komputerów w połowie XX wieku rozpoczął się okres gwałtownego rozwoju społeczeństwa cyfrowego. Elektroniczne maszyny liczące, które zaczęły pojawiać się najpierw w zastosowaniach naukowych, ekonomicznych oraz wojskowych, obecnie wspierają człowieka niemal w każdym rodzaju jego aktywności. Dotyczy to nie tylko wykonywania długich i żmudnych obliczeń, ale również wsparcia narządów zmysłu (np. wzroku). W szczególności, na przestrzeni ostatniego półwiecza nastąpił bardzo szybki rozwój cyfrowej analizy obrazów, co zaowocowało powstaniem systemów wizyjnych, które znajdują zastosowanie w wielu dziedzinach, począwszy od astronomii, geologii, fizyki, poprzez medycynę, biologię i ekologię, aż po archeologię i sztukę. Pomimo tego, że systemy te nie są tak doskonałe jak ludzki narząd wzroku, to mogą znacznie wesprzeć i poszerzyć zakres jego możliwości. Przykładowo – fotografia cyfrowa i analogowa pozwalają na utwalenie obrazów, które widzimy przez ułamek sekundy, umożliwiając tym samym ich późniejszą analizę w sposób bardziej uważny, a co za tym idzie bardziej obiektywny. Teleskopy i mikroskopy poszerzają zakres ludzkiego widzenia, pozwalając na obserwowanie obiektów w różnej skali. Nowoczesne technologie mogą również zrekompensować naturalne ograniczenia ludzkiego oka, które „widzi” tylko w ograniczonym przedziale częstotliwości spektrum elektromagnetycznego (Rys. 1.1). Tym samym mamy możliwość obserwowania zjawisk zachodzących nie tylko w zakresie promieniowania widzialnego, ale również promieniowania gamma i rentgenowskiego, jak również w paśmie podczerwieni i fal radiowych. Własność ta już w XIX wieku pozwoliła W.C. Roentgenowi na zarejestrowanie obrazu kości człowieka za pomocą odkrytego przez niego promieniowania X (zwanego również od jego nazwiska *promieniowaniem rentgenowskim*). W kolejnych latach rozpoczął się intensywny rozwój różnego typu urządzeń obrazowania medycznego, a do momentów przełomowych można zaliczyć [47] w szczególności: zbudowanie mikroskopu elektronowego przez M. Knolla i E. Ruska, opracowanie zasad tomografii komputerowej przez A.M. Cormacka, pierwsze zastosowanie ultradźwięków oraz pierwsze wykorzystanie jądrowego rezonansu magnetycznego. Narzędzia te stanowią obecnie podstawę nowoczesnych metod diagnostyki medycznej.



Rysunek 1.1: Spektrum elektromagnetyczne z zaznaczonym zakresem promieniowania widzialnego (źródło: *Wikipedia*).

Szerokie wykorzystanie systemów wizyjnych generuje ogromną ilość informacji obrazowej. Dotyczy to już nie tylko tak specjalistycznych zastosowań, jak diagnostyka obrazowa, czy też obserwacja nieba nowoczesnymi teleskopami. Dzięki upowszechnieniu się cyfrowych aparatów fotograficznych oraz kamer wideo obraz jest gromadzony niemal nieustannie przez coraz większą liczbę urządzeń. Niestety w większości przypadków jakość pozyskiwanych obrazów nie jest zadowalająca. Problem ten szczególnie dotyczy urządzeń mobilnych, które pomimo coraz większych matryc światłoczułych mają bardzo słabą optykę. Chcąc poddać takie obrazy analizie, niezbędne wydaje się podjęcie odpowiednich kroków mających na celu poprawę ich jakości. Można w tym celu skorzystać z metod cyfrowego przetwarzania obrazu, które pozwalają na usunięcie zakłóceń powstałych w trakcie procesu akwizycji, na ich wyostrenie, a także poprawę kontrastu oraz jasności poszczególnych punktów. Nawet tak przygotowane obrazy mogą okazać się bezwartościową informacją, jeśli nie zostaną odpowiednio zinterpretowane. Angażowanie do tej roli tylko i wyłącznie człowieka, nawet jeśli na początku wydaje się to najlepszym rozwiązaniem, może powodować szereg problemów, które wynikają przede wszystkim z ograniczeń ludzkiego narządu wzroku. Pomimo tego, że ludzki mózg przetwarza dane obrazowe z prędkością ok.  $100MB/s$  [69], to jednak szybko ulega znużeniu, jeśli obserwowana jest przez dłuższy czas ta sama scena. Dodatkowo nasze oko reaguje raczej na zmiany wartości bodźców, niż na ich wartości bezwzględne. Czynniki te znacznie zmniejszają dokładność oceny zjawisk obserwowanych na obrazie i uniemożliwiają uznanie jej za obiektywną. W takiej sytuacji konieczne wydaje się wprowadzenie kolejnych etapów cyfrowej obróbki obrazu, i na ile jest to możliwe, dążenie do pełnej automatyzacji procesu pozyskiwania

i oceny obrazu. Istotną rolę odgrywają tu zaawansowane algorytmy segmentacji obrazu, detekcji i śledzenia obiektów, a także metody rozpoznawania wzorców. Algorytmy te ułatwiają poszukiwanie i mierzenie analizowanych obiektów, ale nie są w stanie dokonać pełnej interpretacji zjawisk zarejestrowanych w obrazie. W tym celu niezbędna jest specjalistyczna, często również interdyscyplinarna wiedza, którą posiada osoba dokładnie znająca badane zagadnienie. Z taką sytuacją spotykamy się np. w medycynie, gdzie lekarz bazując na swojej wiedzy i doświadczeniu potrafi ocenić obraz i na tej podstawie dokonuje diagnozy stanu zdrowia pacjenta. Właśnie taka wiedza i doświadczenie, po odpowiednim zamodelowaniu w algorytmach przetwarzania obrazu, pozwala na prowadzenie prac naukowych [34, 24, 32] ukierunkowanych na tworzenie automatycznych lub półautomatycznych metod analizy obrazu. Powstają również prace [49, 70], których celem jest transfer wiedzy eksperckiej do systemów komputerowych w celu umożliwienia automatycznego rozumienia obrazów. Tego typu badania dają nadzieję, że w niedalekiej przyszłości nowoczesne systemy wizyjne zdobędą uznanie w kolejnych dziedzinach życia jako realne wsparcie dla człowieka.

W chwili obecnej wyzwaniem dla inżynierów tworzących systemy wizyjne jest konstruowanie uogólnionych algorytmów, które będą mogły znaleźć zastosowanie w wielu różnych przypadkach, a nie będących wyłącznie rozwiązaniami dedykowanymi dla konkretnego problemu. Znaczna część popularnych algorytmów segmentacji obrazu pozwala jedynie wykryć elementarne obiekty występujące na obrazie, a nie całe obszary, które odpowiadają obiektom istniejącym w rzeczywistości. Podejście to jest sprzeczne z naturą zmysłu wzroku, który najpierw koncentruje naszą uwagę na głównych obiektach, które widzimy, a dopiero później pozwala nam dostrzec ich szczegóły. W niniejszej pracy zostanie podjęta tematyka detekcji takich obiektów, zwanych dalej **dużymi obiektami**.

Ścisłe zdefiniowanie dużego obiektu jako elementu obrazu napotyka na podstawową trudność, wynikającą z faktu, że obraz stanowi jedynie pewne odwzorowanie rzeczywistości. Wyobraźmy sobie pojedynczą osobę stojącą przed gładką ścianą. Jeśli w odpowiedni sposób oświetlimy tą osobę, to na ścianie uzyskamy jej cień. Sfotografowanie tego cienia doprowadzi do powstania obrazu, na którym z łatwością wyróżnimy jeden duży obiekt, a mianowicie cień omawianej osoby. Jeżeli zrobilibyśmy zdjęcie tej osoby i wszystkie punkty jej obrazu istotnie różniłyby się od tła (będzie tak zapewne np. w przypadku osoby w ciemnym ubraniu i śnieżnobiałej ścianie), to również względnie łatwe będzie jej rozpoznanie jako jednego dużego obiektu. Sprawa skomplikuje się, gdy sfotografujemy osobę ubraną na biało lub dwie osoby, szczególnie, gdy będą np. przytulone do siebie i ubrane w podobny sposób. Obserwator takiego wydarzenia prawdopodobnie nie będzie miał problemów z rozróżnieniem osób, o których mowa, jednak zadanie to może być już

dosyć skomplikowane do rozwiązania z pomocą ogólnie znanych metod komputerowej analizy obrazu.

Podstawą zaliczenia danej grupy pikseli jako obiektu powinno być to, że odzwierciedlają one obiekt istniejący fizycznie, a nie fakt posiadania wspólnej cechy, takiej jak kolor, poziom szarości, czy sąsiedztwo. W ogólnym przypadku do detekcji tak rozumianych dużych obiektów nie wystarczy ocena wartości poszczególnych punktów – konieczne jest uwzględnienie wiedzy o fizycznych cechach badanych obiektów. Zapis takiej wiedzy w postaci sformalizowanej może być bardzo trudny, szczególnie, jeśli założymy, że akwizycja obrazu może przebiegać nie tylko za pomocą powszechnie znanych aparatów fotograficznych, ale również innych urządzeń, takich jak kamera termowizyjna, aparat rentgenowski lub tomograf komputerowy.

Duże obiekty w rozumieniu niniejszej pracy, tak naprawdę nie muszą być w rzeczywistości obiektami o dużych rozmiarach. Ważne jest to, aby były to obiekty duże w skali obrazu. Może to być np. całe osiedle mieszkalne lub seria blisko rosnących, nachodzących na siebie drzew. Z drugiej strony może to być np. pojedyncza żywa komórka, pod warunkiem, że zajmuje znaczny obszar przedstawiającego ją obrazu. Obiektywnie trudno wskazać wartość progową, która oddzielałaby duże obiekty od małych, ale można zaproponować kilka kryteriów ułatwiających takie rozróżnienie.

1. Wielkość obiektu względem innych obiektów. Kryterium to można rozpatrzyć na dwa sposoby:

- Jeśli obraz przedstawia podobne obiekty, to można założyć, że będą one dużymi obiektami, jeśli razem zajmują większą część obrazu, a ich liczba nie przekracza kilku (maksymalnie kilkunastu) egzemplarzy. Przykładem opisywanej tutaj sytuacji może być obraz mocno powiększonej struktury komórkowej.
- W przypadku obrazu, na którym występują różne obiekty, do dużych można zaliczyć te o największym rozmiarze, pod warunkiem, że w sumie ich powierzchnia również zajmuje większą część obrazu, a liczebność nie przekracza kilku, względnie kilkunastu sztuk.

2. Wizualna odrębność obiektu. Duży obiekt na obrazie niekoniecznie musi być obiektem pojedynczym. Może on składać się z mniejszych, składowych obiektów. Taką grupę można sklasyfikować jako jeden obiekt, jeśli istnieje problem z wizualnym oddzieleniem poszczególnych obiektów od siebie. Przykładem może być grupa drzew rosnących blisko obok siebie, które na siebie nachodzą. Dodatkowo, sklasyfikowaniu zbioru obiektów jako jeden duży obiekt może sprzyjać liczba tych obiektów. Jeśli jest ich kilkadziesiąt, a nawet kilkaset, a do tego tworzą one skupisko oddzielone od innych obiektów, mogą być one uznane jako jeden duży obiekt. Jako przykład

można podać tłum ludzi na koncercie muzycznym stojących na otwartej przestrzeni. W tym przypadku kryterium klasyfikacji może być np. stosunek gabarytów obiektu do jego odległości od najbliższych sąsiadów.

Podane kryteria nie są kryteriami ściśle liczbowymi, ale raczej zbiorem wskazówek i cech dużych obiektów, które mogą ułatwić ich rozróżnienie spośród innych obiektów na obrazie. Kryteria te mogłyby zostać opisane za pomocą teorii liczb i zbiorów rozmytych, jednak różnorodność sytuacji, jakie mogą zostać przedstawione na obrazie może być tak duża, że bardzo trudne, wręcz niemożliwe jest jednoznaczne określenie przedziałów klasyfikacji. W związku z tym w dalszej części pracy zakłada się, że każda seria zdjęć, która będzie podlegała analizie, powinna zostać wcześniej oceniona przez użytkownika, czy zawiera duże obiekty czy też nie. Taka ocena, pomimo tego, że będzie oceną subiektywną, jest niezbędna, aby uszczegółowić jaką przestrzeń na obrazie mają zajmować duże obiekty i jaka może być ich maksymalna liczebność (*kryterium 1*) i czy jest możliwe wizualne rozróżnienie poszczególnych elementów tworzących większą grupę (*kryterium 2*).

Wstępna ocena zdjęć wykonana przez użytkownika, może być również niezbędna tam, gdzie jest potrzebna dodatkowa wiedza pozwalająca określić granice zalegania dużych obiektów. Wiedza ta może wynikać z doświadczenia, a nawet intuicji osoby oceniającej, co jeszcze bardziej utrudnia sformalizowanie jej w postaci przedziałów i kryteriów liczbowych. Aby zilustrować tego typu sytuację, można wyobrazić sobie zdjęcie lotnicze lub satelitarne przedstawiające duże miasto (przykłady na rysunkach 1.2a i 1.2b). Jeśli zadaniem byłoby wyznaczenie obszaru miasta na tym obrazie, to można wskazać kilka przykładowych problemów z tym związanych:

- obszar centrum (wysoka zabudowa, mało zieleni) różni się znacznie od przedmieść i dzielnic willowych,
- dzielnice willowe z kolei wyglądają podobnie, jak otaczające miasto mniejsze miasteczka, które do niego nie należą,
- część miasta jest zacieniona lub wręcz zasłonięta przez chmury, co utrudnia lub uniemożliwia rozpoznanie niektórych obszarów,
- w przypadku gdy miasto położone jest w terenie górzystym, zabudowa może być znacznie rzadsza i porośnięta dodatkowo licznymi drzewami i krzewami upodabniając się tym samym do licznych podmiejskich (już poza miastem) nieużytków.

Wskazane powyżej problemy pomimo pewnego stopnia skomplikowania, wydają się łatwe do rozwiązania w drodze manualnego zaznaczenia obszaru miasta przez użytkownika





(a)



(b)

Rysunek 1.2: Zdjęcia przykładowych miast ilustrujące problemy detekcji dużych obiektów.

(a) Bergen, Norwegia (autor: *Leszek Wojnar*).

(b) Salt Lake City, Utah, USA (autor: *Rick Willoughby*).

na obrazie. To samo zadanie, jeśli miałyby zostać rozwiązane w pełni automatycznie przez komputer, nie wydaje się już takie trywialne. Potrzeba podjęcia tego zadania wynika przede wszystkim z szybkości i powtarzalności analizy wykonywanej przez

komputer. Powtarzalność wyników analizy może z powodzeniem zrównoważyć drobne błędy w detekcji obiektów.

Trudności w analizie dużych obiektów na obrazach komputerowych mogą wynikać nie tylko z problemu obiektywnego oddzielenia ich od pozostałych obiektów (może to wymagać nawet zaangażowania osoby zewnętrznej, która wskaże ich dodatkowe cechy), ale również z problemów z ich akwizycją i cyfrowym zapisem. Nierzadko zdarza się, że z powodu swojej wielkości obiekty te bywają nierównomiernie oświetlone, a ich krawędzie w różnych miejscach obrazu mają różne poziomy ostrości. Często bywa tak, że mają one strukturę niejednorodną i mogą wyglądać na obrazie jako zbiór kilku obiektów, choć w istocie tworzą jeden większy obiekt. W dalszej części pracy zostanie dodatkowo rozpatrzona możliwość występowania zakłóceń w obrazie, które jeszcze bardziej utrudniają proces detekcji dużych obiektów.

Wymienione powyżej przykładowe problemy w istotny sposób utrudniają analizę obrazów przedstawiających duże obiekty. Istnieje wiele typów obrazów, których może to dotyczyć. Oto niektóre z nich:

- wybrane składniki struktur metali i ich stopów (np. obszary eutektyk (Rys. 1.3a)),
- obrazy przedstawiające naturę (Rys. 1.3b i 1.3e) - zarówno rośliny jak i zwierzęta,
- obszar narządów wewnętrznych na obrazach z tomografii komputerowej, np. obszar płuc (Rys. 1.3c),
- ślady zużycia na powierzchni różnych elementów (np. zużyta opona samochodowa (Rys. 1.3d) lub „wyślizgany” obszar touch-pada),
- obszar elementów kostnych, np. obszar kości udowej (Rys. 1.3f) oraz innych obiektów (Rys. 1.3g, 1.3h i 1.3i) na obrazach rentgenowskich.

Dla obrazów takich jak te opisane powyżej i innych, na których występują duże obiekty, celem rozprawy będzie opracowanie nowych metod przetwarzania obrazu, działających podobnie jak nasz zmysł wzroku, który ma zdolność do eliminacji zakłóceń oraz umiejętność korzystania z bazy wiedzy. Cel ten zostanie dokładnie sprecyzowany pod koniec rozdziału 2, gdzie została rozwinięta problematyka detekcji dużych obiektów oraz został przedstawiony aktualny stan badań w tym zakresie.



Rysunek 1.3: Obrazy prezentujące przykłady dużych obiektów (dokładny opis źródeł w tekście; Rys. 1.3a i 1.3d źródło Internet. Rys. 1.3b i 1.3e pobrano z ogólnodostępnej bazy obrazów Berkeley Segmentation Dataset and Benchmark [44]. Rys. 1.3c jeden z przykładowych obrazów płuc pozyskanych z Collegium Medicum Uniwersytetu Jagiellońskiego. Rys. 1.3f jeden z obrazów radiologicznych kości udowej pozyskany z Krakowskiego Centrum Medycznego. Rys. 1.3g, 1.3h i 1.3i pochodzą internetowej bazy obrazów medycznych Image Retrieval in Medical Applications).

Rozprawa została podzielona na 5 części:

- I **Wprowadzenie** poza ogólnym wstępem do problematyki detekcji dużych obiektów (Rozdz. 1), zawiera bardziej szczegółową analizę podejmowanego tematu pracy przy uwzględnieniu aktualnego stanu wiedzy (Rozdz. 2). Pod koniec rozdziału „*Analiza problemu, cel i zakres pracy*” określono cel pracy, podano jej zakres oraz postawiono tezy, które będą weryfikowane w kolejnych częściach pracy.
- II **Przygotowanie obrazu do dalszej analizy** przedstawia metody, których użycie jest zalecane na etapie wstępnej obróbki materiału badawczego przed przystąpieniem do właściwego procesu jego analizy. Omówiono tutaj metody usuwania szumu z obrazu (Rozdz. 3), algorytmy wstępnej segmentacji (Rozdz. 4) oraz znane techniki łączenia regionów (Rozdz. 5), które były testowane przed opracowaniem własnych algorytmów.
- III **Algorytmy segmentacji oraz detekcji dużych obiektów** stanowi główną część pracy, gdzie zaprezentowano trzy koncepcje segmentacji i detekcji dużych obiektów. Każda z tych koncepcji (Rozdz. 6, 7 i 8) bazuje na powszechnie znanych w literaturze metodach przetwarzania obrazu, a wkładem własnym autora niniejszej pracy są dwa nowe algorytmy (Rozdz. 6.2 i 7.2), których celem jest usprawnienie ich działania i poprawa jakości uzyskiwanych przez nie wyników.
- IV **Podsumowanie** zawiera nie tylko końcowe wnioski (Rozdz. 10), ale również wskazuje na propozycje wykorzystania opracowanych metod w systemie automatycznego rozpoznawania i detekcji dużych obiektów, którego projekt i etapy realizacji opisano w rozdziale 9.
- V **Dodatki**. Ostatnia część pracy zawiera opis znanych w literaturze algorytmów analizy obrazu (Dodatek A i B), które były podstawą opracowania własnych metod badawczych. Pomimo tego, że prawdopodobnie znane są one osobom na co dzień zajmującym się komputerową analizą obrazu, to zostały dołączone, aby pozostałe osoby również mogły poznać zasady ich działania.



W celu graficznej ilustracji podejmowanej tematyki, a następnie opisu przygotowania danych do analizy i ostatecznej prezentacji działania opracowanych metod wykorzystano wiele przykładowych obrazów. Pochodzą one nie tylko z publicznie dostępnych zasobów Internetu, ale zostały również udostępnione przez lekarzy z krakowskich ośrodków medycznych. Poniżej przedstawiono krótką charakterystykę tych danych.

- Obrazy radiologiczne kości udowej (przykładowy obraz na rysunku 1.3f) - seria 20 obrazów rentgenowskich przedstawiających lewą kość udową udostępnione przez *Krakowskie Centrum Medyczne*<sup>1</sup>.
- Obrazy tomograficzne klatki piersiowej (przykładowy obraz na rysunku 1.3c) - seria 25 skanów tomograficznych wykonanych dla 5 różnych pacjentów chorych na astmę (5 skanów dla jednego pacjenta). Obrazy pochodzą z *II Katedry Chorób Wewnętrznych Collegium Medicum Uniwersytetu Jagiellońskiego*<sup>2</sup>.
- Obrazy z projektu *Image Retrieval in Medical Applications - IRMA 2009*<sup>3</sup>, do których uzyskano licencję na ich przetwarzanie na potrzeby badań naukowych z Zakładu Informatyki Medycznej Uniwersytetu Aachen. Przykładowe obrazy pokazano na rysunku 1.3g, 1.3g i 1.3g.
- Obrazy z publicznie dostępnej bazy danych *Berkeley Segmentation Dataset and Benchmark*<sup>4</sup> (przykładowy obraz na rysunku 1.3b i 1.3e), która została przedstawiona w artykule [44]. Baza ta jest często wykorzystywana do testowania i porównywania wyników działania nowo tworzonych algorytmów detekcji krawędzi w obrazie.

Dla pozostałych obrazów, nie wymienionych powyżej, ich źródła pochodzenia podano każdorazowo w momencie ich użycia w tekście.

---

<sup>1</sup>Krakowskie Centrum Medyczne, ul. Mikołaja Kopernika 32, Kraków.

<sup>2</sup>II Katedra Chorób Wewnętrznych CM UJ, ul. Skawińska 8, Kraków.

<sup>3</sup>Więcej informacji o bazie danych *Image Retrieval in Medical Applications 2009* można przeczytać na stronie <http://ganymed.imib.rwth-aachen.de/irma/>.

<sup>4</sup>Baza danych jest dostępna w Internecie pod następującym adresem: <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>.

## Rozdział 2

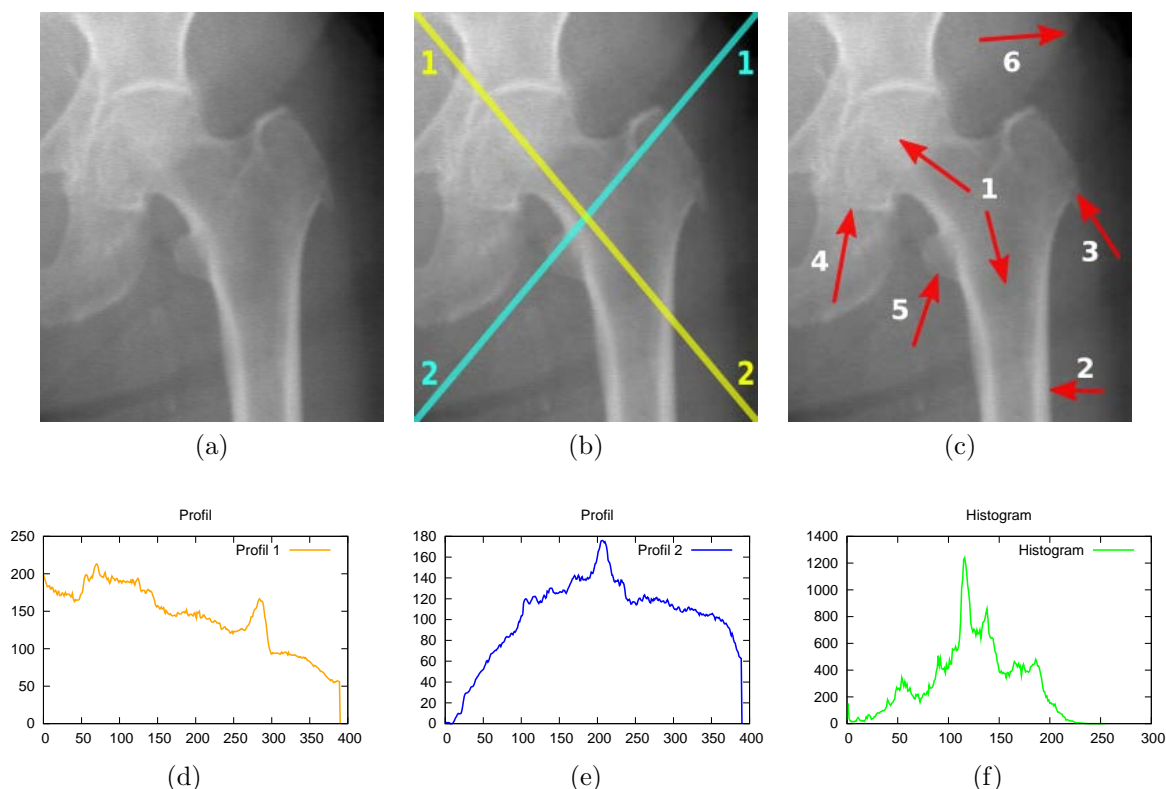
---

### *Analiza problemu, cel i zakres pracy*

Standardowy proces działania komputerowych metod analizy obrazu składa się z kilku etapów [75]. Najpierw z obrazu usuwany jest szum wynikający z niedoskonałości aparatury do akwizycji obrazu (etap 1). Następnie wyznaczana jest granica zalegania obiektu lub grupy obiektów, które w danym momencie są przedmiotem badań (etap 2). Ostatecznie obiekty są mierzone (etap 3), a wynik jest prezentowany użytkownikowi lub zapisywany do bazy danych (ewentualnie pliku lub arkusza kalkulacyjnego). Dodatkowo program może dokonać klasyfikacji danego obiektu, jeśli zostały zdefiniowane odpowiednie kryteria oceny (etap 4). Zarówno dla pierwszego, jak i trzeciego etapu istnieje wiele metod ogólnych, które sprawdzają się w zdecydowanej większości przypadków. Algorytmy usuwania szumu najczęściej starają się wygładzić obraz biorąc pod uwagę otoczenie punktu. Stosując proste obliczenia statystyczne, takie jak mediana lub średnia, można szybko usunąć wartości skrajne wprowadzane przez szum. Z kolei pomiary wykonywane są najczęściej na obrazie binarnym, przedstawiającym poszukiwane obiekty i obejmują wyznaczenie różnych parametrów geometrycznych i statystycznych. Etapem, od którego najsilniej zależy powodzenie całego procesu komputerowej analizy obrazu jest detekcja obiektów. Często jest to zadanie na tyle trudne, że wymaga specjalistycznej wiedzy z zakresu przetwarzania obrazów. Projektowane algorytmy są albo zbyt ogólne tak, iż nie są w stanie rozwiązać postawionego przed nimi zadania w stopniu satysfakcjonującym, albo są dedykowane do konkretnego zbioru obrazów, przez co nie jest możliwe ich zastosowanie dla innych typów obrazów. Wstępnym zadaniem jakie postawiono sobie na początku rozpoczęcia prac nad niniejszą rozprawą, było **umożliwienie łatwej konfiguracji parametrów algorytmu detekcji, tak aby mógł on zostać dostosowany do analizy nie tylko jednego, ale wielu różnych grup obiektów**. Ważną kwestią, którą brano również pod uwagę jest możliwość **ukrycia przed użytkownikiem szczegółów związanych z metodami komputerowej analizy obrazu i wykorzystanie przede wszystkim jego wiedzy w zakresie obejmującym badanie**.

Proponowany zakres rozprawy doktorskiej dotyczy problemu detekcji obiektów, które z powodu swojej wielkości oraz ewentualnego zróżnicowania zarówno tła, jak i samych obiektów, nie mogą być wydzielone z obrazu na podstawie prostych, jednoznacznych parametrów. Może to być związane z lokalnymi własnościami obiektu (np.

lokalne minima/maksima), losowymi zaburzeniami (przysłonięcie innym obiektem) lub globalnymi własnościami obrazu (np. nierównomierne oświetlenie). Dodatkowy problem mogą stwarzać nieostre granice obszaru, który płynnie przechodzi w tło lub łączy się z innymi elementami. Rozważone zostanie również występowanie szumu powodującego zakłócenia poziomu szarości lub koloru na poziomie pojedynczych pikseli obrazu. Powyższe problemy dobrze ilustruje radiogram kości udowej pokazany na rysunku 2.1a.



Rysunek 2.1: Wskazanie problemów w analizie dużych obiektów na przykładzie obrazu (a) kości udowej. (b) Obraz z nałożonymi liniami, dla których wygenerowano profile widoczne na (d) i (e). Profil (d) odpowiada linii żółtej na obrazie (b), natomiast profil (e) linii niebieskiej. Kierunek profilu został określony cyframi zaznaczonymi na (b). Obraz (c) wskazuje na problematyczne fragmenty obrazu z punktu widzenia jego automatycznej analizy (*szczegółowy opis w tekście*). Wykres (f) przedstawia wykres histogramu dla (a). (*Opracowanie własne*).

Na obrazie tym można wyróżnić kilka elementów: obszar miednicy, obszar kości udowej oraz tło. Kość udowa pomimo tego, że jest ściśle wpasowana w miednicę, anatomicznie jest osobnym elementem kostnym. Dalsze cechy anatomiczne nie będą analizowane, zostaną natomiast wskazane problemy występujące na tym obrazie z punktu widzenia jego cyfrowego przetwarzania, a w szczególności wydzielenia obszaru kości udowej. Na początku warto prześledzić informacje jakie daje wygenerowanie histogramu oraz profili dla wybranych linii. Histogram pokazany na rysunku 2.1f jest rozkładem wartości pikseli

w obrazie. Nie wskazuje on tutaj wyraźnego rozgraniczenia pomiędzy poszczególnymi obszarami obrazu, co może utrudniać wyznaczenie progu binaryzacji. Podobne wnioski można wysunąć obserwując profile wyznaczone na przekątnych obrazu. Widać w nich nie tylko brak występowania ostrych granic pomiędzy obszarem kości udowej a miednicą oraz obszarem kości udowej a tłem obrazu, ale również zmienny poziom wartości pikseli w obrębie samej kości udowej (wykres 2.1d). Te i inne problemy dodatkowo zaznaczono strzałkami na rysunku 2.1c. Wskazują one na następujące cechy, które mogą być problematyczne podczas przetwarzania tego obrazu:

- nierównomierny poziom jasności pikseli w obrębie kości udowej (strzałka 1),
- zmieniająca się ostrość granicy pomiędzy kością udową a tłem, począwszy od wyraźnie zaznaczonej w dolnej części obrazu (strzałka 2), po słabszą w jego środkowej części (strzałka 3),
- brak wyraźnej granicy pomiędzy kością udową a miednicą (strzałka 4),
- dodatkowe artefakty przylegające do kości udowej (strzałka 5), które powinny zostać wskazane jako osobny obiekt w celu późniejszej analizy,
- nierównomierne tło (strzałka 6).

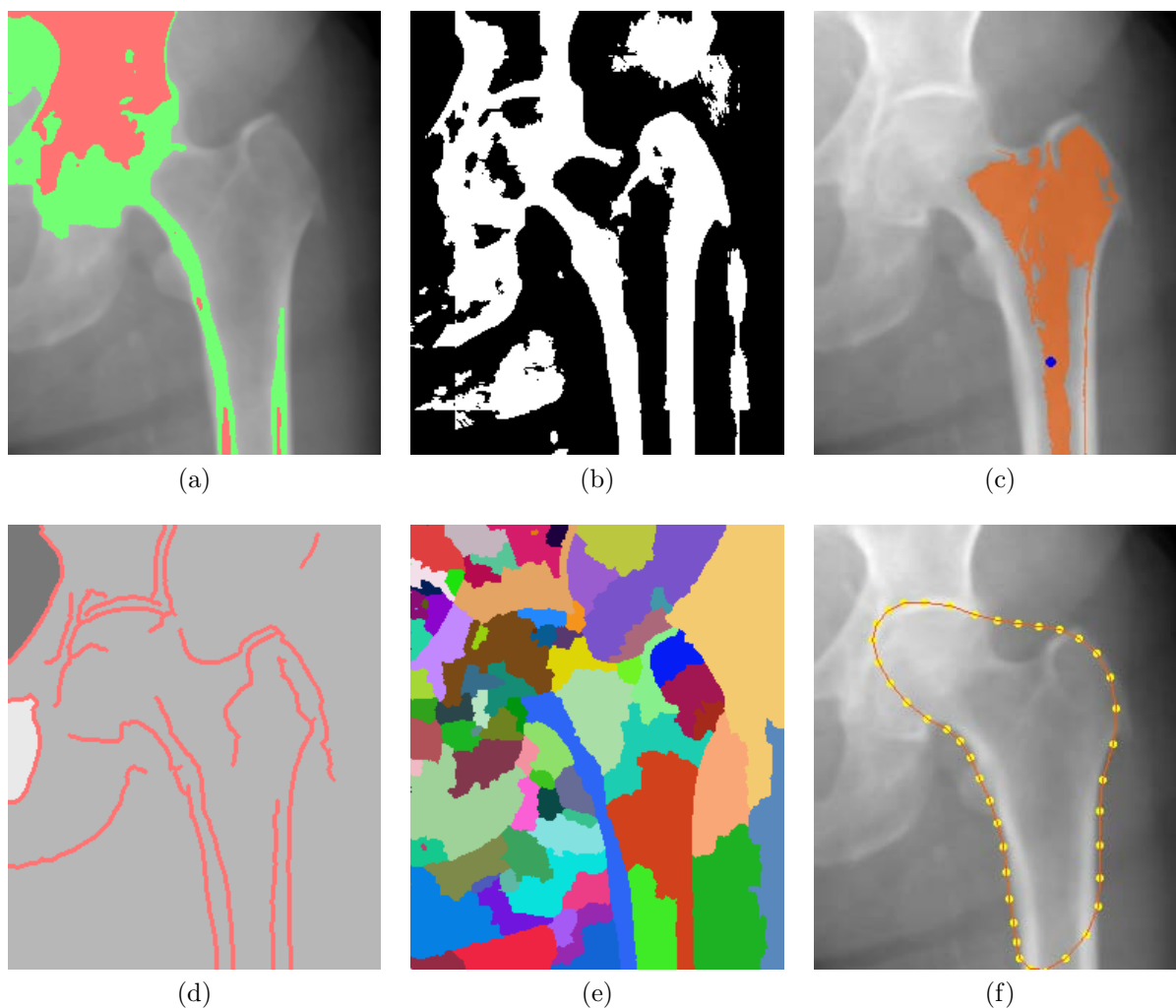
Tego typu utrudnienia komplikują przygotowanie algorytmu detekcji badanego obiektu, niezależnie od tego, czy jest to obraz przedstawiający kość udową, czy też dowolny inny obraz. Duże obiekty, które w nich występują uniemożliwiają zastosowanie jednego algorytmu segmentacji i wymagają konstruowania bardziej złożonych, często dedykowanych algorytmów. Aby pokazać problemy, jakie wiążą się z komputerową interpretacją obrazu dla dużych i niejednorodnych obiektów przeanalizowano grupę przykładowych obrazów<sup>1</sup> z wykorzystaniem klasycznych algorytmów segmentacji. Wnioski z tej analizy przedstawiono poniżej:

- Algorytmy progowania, związane z poszukiwaniem optymalnej wartości granicznej [51, 77] oddzielającej obiekt od tła lub od innych obiektów, nie działają najlepiej. Znajdowana przez nie wartość (próg binaryzacji) potrafi poprawnie wskazać tylko część obiektu, dodając lub pomijając inne elementy. Ponadto, różne obiekty na tym samym obrazie mogą wymagać różnych wartości progowych. Również zastosowanie progowania z histerezą, dla którego określa się dwie wartości progu - wyższą, która zaznacza fragmenty obiektu, oraz niższą, której zadaniem

---

<sup>1</sup>Testy prowadzone były z wykorzystaniem obrazów wskazanych w poprzednim rozdziale niniejszej pracy, gdzie opisano ich źródła pochodzenia. Omawiane algorytmy będą ilustrowane przede wszystkim za pomocą obrazu kości udowej, dla którego pokazano podstawowe problemy z punktu widzenia jego komputerowej obróbki.





Rysunek 2.2: Próby segmentacji obiektów na obrazie przedstawiającym fragment miednicy oraz kości udowej z wykorzystaniem standardowych metod przetwarzania obrazu (*dotatkowy opis w tekście*).

(a) Progowanie z histerezą z progiem wyższym równym 204 (obszar czerwonym) i progiem niższym równym 178 (obszar zielony). Jak widać na tym rysunku, szybciej zostały wliczone punkty należące do tła (lewy górny róg), aniżeli punkty należące do obiektu.

(b) Rezultaty progowania adaptacyjnego o wartości promienia równej 30.

(c) Wyniki rozrostu obszaru. Pomarańczowy obszar uzyskano poprzez przyłączenie wszystkich punktów mających wartość niższą od 35 na obrazie gradientowym. Zwiększenie tej wartości o 1 powoduje rozlanie się obszaru nie tylko na kolejne fragmenty kości, ale również na znaczne fragmenty tła.

(d) Segmentacja bazująca na krawędzi, gdzie wszystkie punkty zamknięte jedną linią krawędzi stanowią osobne obiekty. Z powodu braku ciągłości linii krawędzi na obrazku (d) widzimy tylko 3 segmenty. Detekcję krawędzi wykonano przy użyciu *algorytmu Canny* [7].

(e) Wyniki segmentacji algorytmem działów wodnych, po jego wcześniejszym wygładzeniu filtrem Gaussa o wartości *sigma* równej 1.

(f) Próba wyznaczenia obszaru kości udowej przy użyciu algorytmu aktywnego konturu. (*Opracowanie własne*).

jest określenie pozostałych fragmentów należących do obiektu i przylegających do obszaru uzyskanego wcześniej z wykorzystaniem wartości wyższej - nie sprawdza się na tego typu obrazach (przykład na rysunku 2.2a). Przetestowano również algorytm progowania adaptacyjnego, w którym próg jest wyznaczany dynamicznie na podstawie lokalnego otoczenia (zazwyczaj jest to kwadrat, bądź też okrąg, gdzie długość boku/promienia jest podana przez użytkownika jako parametr wejściowy dla algorytmu). To ostatnie podejście daje podobny rezultat jak na rysunku 2.2b, co znacznie odbiega od oczekiwanego celu analizy.

- Z powodu braku jednorodności obiektu algorytmy rozrostu [20, 1, 22] (ang. *region growing*) startujące z losowych punktów, bądź też punktów wyznaczonych przez użytkownika, zatrzymują się po objęciu tylko części obszaru lub wykrywają zbyt duże obszary (przykład na rysunku 2.2c).
- Algorytmy oparte o detekcję krawędzi z jednej strony zatrzymują się na drobnych krawędziach w obiekcie, a z drugiej strony stają się bezużyteczne w przypadku nieostrych granic. Największym problemem jest jednak fakt braku ciągłości granic i automatyczny dobór tych fragmentów, które mają być sklejone do postaci jednego obiektu<sup>2</sup> (przykład na rysunku 2.2d).
- Wadą dwóch kolejnych narzędzi jest niepotrzebny podział dużego obiektu na kilka mniejszych. Pierwsze z nich – działy wodne<sup>3</sup> [73, 4] (ang. *watershed*) – jest wrażliwe na drobne lokalne gradienty (przykład na rysunku 2.2e), natomiast drugi – dzielenie i łączenie regionów<sup>4</sup> (ang. *split and merge technique*) – zatrzymuje się na pewnym etapie działania nie potrafiąc scalić dwóch niejednorodnych elementów (przykład na rysunku 2.3b).
- Aktywne kontury [2, 42, 50] są narzędziami skutecznymi, ale zwykle wymagają pracy w trybie interaktywnym i są efektywne jedynie w ograniczonym zakresie przypadków (przykład na rysunku 2.2f).

Pomimo tego, że wskazane algorytmy nie dają satysfakcjonujących rezultatów w przypadku ich samodzielnego stosowania, to często bywają używane jako jeden z elementów bardziej złożonych algorytmów. Powstają również liczne ich modyfikacje, które lepiej sprawdzają się od pierwotnych wersji w ściśle określonych sytuacjach

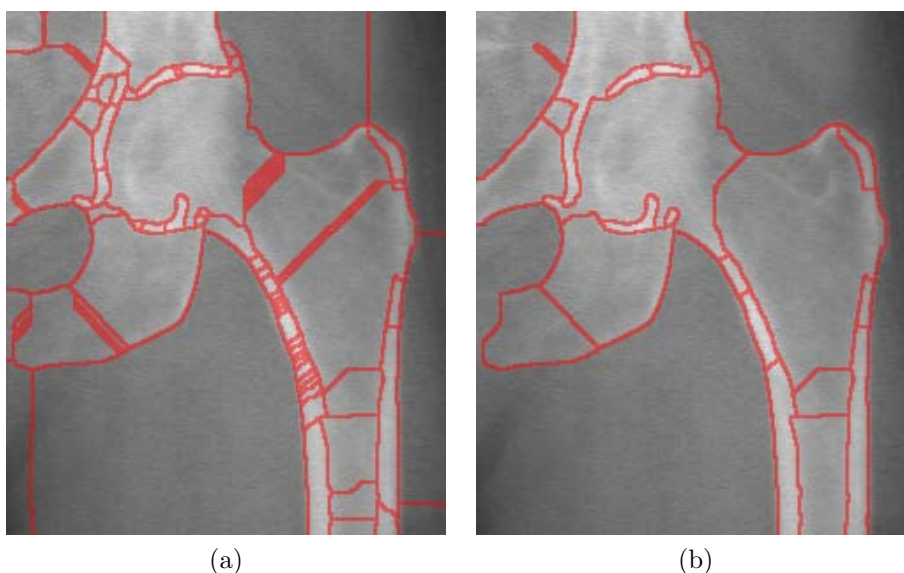
<sup>2</sup>Problematykę tą szerzej opisano w rozdziale 6.1. [Problem nieciągłości krawędzi i sposoby ich łączenia.](#)

<sup>3</sup>Podstawy działania algorytmu działów wodnych opisano w rozdziale 4.1. [Klasyczny algorytm działów wodnych](#) oraz 8.1. [Algorytm działów wodnych bazujący na znacznikach.](#)

<sup>4</sup>Możliwości stosowania algorytmów łączenia segmentów zostały omówione w rozdziale 5. [Łączenie segmentów.](#)

i mogą wymagać spełnienia dodatkowych warunków, np. związanych z przygotowaniem oświetlenia lub wyborem urządzenia akwizycyjnego.

Przeprowadzona analiza literatury pozwala na wskazanie szeregu metod i koncepcji, które mogą znaleźć zastosowanie do detekcji dużych obiektów. Można do nich zaliczyć wymieniony wcześniej algorytm działów wodnych [73, 4], algorytmy aktywnego konturu [2, 42, 50], algorytmy bazujące na teorii grafów [18, 63], a także algorytmy, których podstawą są równania różniczkowe cząstkowe, takie jak *metoda poziomic* (ang. *level set*) [2, 42, 50] i inne. Algorytmy te, nawet jeśli są znacznie bardziej złożone od wcześniej opisanych klasycznych algorytmów, mogą sprawiać problemy w detekcji dużych obiektów, takich jak kość udowa pokazana na rysunku 2.1a. Wynika to z faktu, iż większość z nich traktuje obraz jedynie jako macierz liczb, nie biorąc pod uwagę informacji w nim zawartych i cech wyróżniających badane obiekty od innych obiektów i od tła.



Rysunek 2.3: Możliwości redukcji liczby segmentów poprzez zastosowanie algorytmu łączącego segmenty. (a) Wstępna segmentacja obrazu algorytmem działów wodnych. (b) Wyniki łączenia segmentów. (*opracowanie własne*)

Kolejna grupa algorytmów uwzględnia wiedzę przekazaną przez użytkownika lub zebraną w trakcie wcześniejszego działania (algorytmy bazujące na doświadczeniu). Najprostszą formą tego podejścia, które jest obecnie szeroko stosowane, jest *algorytm łączenia regionów* [28, 62, 71]. Główną ideą jego działania jest scalanie obiektów, które spełniają wcześniej ustalone kryterium jednorodności, będące zazwyczaj pewną miarą statystyczną (np. średnia wartości wszystkich pikseli w dwóch obszarach jest zbliżona, zatem zostaną one połączone (przykład na rysunku 2.3)). Trudnością w ich stosowaniu bywa właściwe określenie kryterium podobieństwa obiektów i wartości progowej mówiącej, które elementy powinny zostać ze sobą połączone. Nawet jeśli uda nam się dobrać te

parametry w odpowiedni sposób, to może się zdarzyć, że wcześniej zostanie włączony inny obiekt, aniżeli ten, o który nam chodziło (problemy tego typu może sprawić np. detekcja osoby w zielonym swetrze stojącej na trawie).

Innym podejściem jest zautomatyzowanie procesu generowania dedykowanych algorytmów dla wcześniej określonych grup obrazów. W tym przypadku podejmowane są próby [11, 56] opisanie wiedzy specjalisty z zakresu przetwarzania obrazu i umieszczenie jej w systemie ekspertowym. Taki system powinien po uprzednim szczegółowym opisanie zadania do analizy wygenerować sekwencję przekształceń obrazu dążącą do wydzielenia poszukiwanego obiektu. Jednak z powodu trudności ze sformalizowaniem wiedzy ekspertów metody te ciągle pozostają w sferze koncepcji i nie znajdują szerokiego zastosowania praktycznego.

Metody inżynierii wiedzy w analizie obrazu są wykorzystywane nie tylko w kontekście systemów ekspertowych, ale również używa się ich do przygotowania obszernych baz danych wzorców z danej dziedziny [41, 61, 5]. Zakłada się, że dla każdego obiektu opisywanego w bazie danych powinien zostać zgromadzony odpowiednio duży zbiór przykładów mówiących o tym, jak ten obiekt wygląda i jakie obiekty zazwyczaj go otaczają. Jako przykład można podać zdjęcia przedstawiające miasto z perspektywy przechodnia. Prawdopodobnie znajdą się na nim budynki, ulice, samochody i inne osoby. Zawarty w bazie danych bogaty zestaw wzorców opisujących te obiekty powinien umożliwić poprawną segmentację obrazu i automatyczne nazwanie wydzielonych elementów. Niestety, nawet po zdefiniowaniu dokładnych baz wzorców często występują problemy, kiedy widoczny jest tylko fragment obiektu lub pojawi się inny obiekt podobny do niego.

Opisane powyżej podejście bywa również stosowane na potrzeby diagnostyki medycznej i nosi nazwę *segmentacji bazującej na atlasie* [59, 52, 58, 37] (ang. *Atlas based image segmentation*). Sprawdza się ono w przypadku obrazów bardzo zbliżonych do siebie, np. sąsiadujących przekrojów z gęsto wykonanych zdjęć tomograficznych, gdzie wskazanie na jednym obrazie badanego obiektu pozwala na jego automatyczne wyznaczenie na pozostałych obrazach. Natomiast w sytuacji, gdy występują znaczne różnice w kształcie analizowanych obiektów, wyniki te bywają na tyle niedokładne, iż uniemożliwiają wykonywanie precyzyjnych pomiarów. W związku z tym mogą one być stosowane jedynie jako pomoc przy wizualnej ocenie obiektów.

Inną metodą, która ma podobny sposób działania są *aktywne modele kształtu* (ang. *Active shape models - ASM*) [13, 12]. Tutaj, w odróżnieniu od metod bazujących na atlasie, gdzie kolejne obrazy są porównywane do obrazu wzorcowego, poszukiwany obiekt jest opisywany za pomocą zbioru wektorów charakteryzujących jego kształt. Wprowadzając proces uczenia algorytmu z wykorzystaniem pewnego wstępnego zbioru obrazów,

możliwe jest dodatkowo wyznaczenie informacji o najbardziej charakterystycznych modyfikacjach kształtu analizowanego obiektu. Na dalszym etapie analizy taki model może być modyfikowany przez dodatkowe algorytmy, które starają się dopasować go do rzeczywistego kształtu, nie dopuszczając jednocześnie do nienaturalnych deformacji. Opisane w pracach [66] i [65] możliwości zastosowania tego typu algorytmów do lokalizacji twarzy i do lokalizacji pojazdów wskazują na szerokie ich zastosowanie, przy czym występują problemy dotyczące m.in. efektywnego uczenia algorytmu, wyznaczania punktów kontrolnych i dalszego ich dopasowania do obiektu. Utrudniają one w znacznej mierze pełną automatyzację tego typu algorytmów.

### Cel i teza pracy

Analiza dużych obiektów w obrazach o wysokim poziomie szumu i niejednorodności dostarcza wielu problemów. Ich detekcja z wykorzystaniem powszechnie znanych algorytmów segmentacji obrazu nie przynosi pożądanych efektów i często wymaga konstrukcji dedykowanych algorytmów dla każdego przypadku z osobna. Takie podejście wymaga nie tylko specjalistycznej wiedzy z zakresu przetwarzania obrazu, ale często wiąże się z wysokimi kosztami opracowania nowej metody oraz długim czasem wprowadzenia jej na rynek - począwszy od etapu koncepcyjnego, aż po docelowe wdrożenie w sprzęcie lub oprogramowaniu u potencjalnego użytkownika.

W związku z tym jako cel pracy przyjęto **opracowanie metod komputerowej analizy obrazu wspomagających detekcję wybranych obiektów w obrazach o wysokim poziomie szumu i niejednorodności, które bez modyfikacji zawartych w nich algorytmów będzie można dostosować do detekcji różnych obiektów**. Docelowo zaproponowane w pracy metody po etapie wstępnej konfiguracji powinny działać w sposób automatyczny.

W świetle tak sformułowanego celu przyjęto następującą tezę pracy:

*Po etapie odpowiedniej filtracji obrazu, w której zostanie usunięty nie tylko szum, ale również drobne elementy zaburzające strukturę obiektu, możliwe jest przyjęcie takiej strategii postępowania, która pozwala na określenie obszaru zalegania dużych obiektów. Strategia ta bazuje na przygotowanym przez użytkownika modelu obiektu, który zawiera informacje o jego cechach charakterystycznych (np. informację o teksturze, kształcie, granicach oraz położeniu względem innych obiektów). Następnie, poprzez analizę tak przygotowanego modelu na jednym obrazie, możliwe jest wskazanie podobnych obiektów na kolejnych obrazach w sposób automatyczny.*

## Część II

# Przygotowanie obrazu do dalszej analizy

Zanim w pracy zostanie podjęta problematyka detekcji dużych obiektów na obrazach niejednorodnych, zostaną przedstawione niezbędne algorytmy służące do przygotowania obrazu do właściwej analizy. Będą to zarówno proste, jak i bardziej złożone algorytmy usuwania zakłóceń z obrazu, które pozwalają zachować strukturę obiektów z możliwie małym rozmyciem granic między nimi. Omówione będą także popularne metody segmentacji obrazu – algorytm działów wodnych i algorytm grafowy wyznaczający minimalne drzewo rozpinające. Algorytmy te pozwalają na skuteczną detekcję jednorodnych obiektów, które znacznie odróżniają się od innych obiektów i od tła. Niestety, segmentacja dużych obiektów o zmiennej strukturze – zarówno w kontekście jasności pikseli, jak i różnorodności tekstury – powoduje zwykle duże ich rozdrobnienie na wiele mniejszych składowych. Często stosowanym rozwiązaniem dla tego problemu są algorytmy łączenia segmentów. Ich rolą jest redukcja liczby podziałów i tworzenie większych obiektów, które spełniają zadane kryteria łączenia. Niestety dla obrazów niejednorodnych przygotowanie odpowiednich kryteriów jest bardzo trudne i daje tylko częściową poprawę wyniku detekcji.

Metody opisane w tej części pracy będą miały istotny udział w proponowanych w kolejnych rozdziałach koncepcjach detekcji dużych obiektów:

- Algorytmy usuwania szumu (Rozdz. 3) – będą stosowane zawsze przed przystąpieniem do właściwej analizy obrazu. Pozwolą na usunięcie zbędnych elementów z obrazu, a tym samym ułatwią poszukiwanie dużych obiektów.
- Algorytmy wstępnej segmentacji obrazu (Rozdz. 4) – zostaną użyte do wyodrębnienia elementarnych struktur na obrazie. Segmenty przygotowane na tym etapie będą podstawą właściwego procesu detekcji, który zostanie zaproponowany w dalszej części pracy.
- Algorytmy łączenia regionów (Rozdz. 5) – ich rolą będzie łączenie wielu segmentów w obiekty odpowiadające rzeczywistym obiektom na obrazie. Zostaną omówione powszechnie znane kryteria łączenia segmentów i problemy w ich stosowaniu do analizy obrazów niejednorodnych.

## Rozdział 3

---

### *Metody usuwania szumu*

Jednym z istotnych czynników, który wpływa na powodzenie procesu analizy obrazu jest jego jakość. Pożądana byłaby taka jakość obrazu wejściowego, która wiernie oddaje rzeczywistą strukturę badanego obiektu. Niestety, wiele czynników wpływa na to, że ten ideał jest trudny do osiągnięcia, co objawia się niską ostrością obrazu, lub wysokim poziomem zakłóceń. Do głównych czynników, które negatywnie wpływają na jakość obrazu można zaliczyć:

- niedoskonałość urządzeń akwizycyjnych wynikającą z aktualnego stanu techniki lub też zewnętrznych ograniczeń (np. dopuszczalnego poziomu promieniowania w badaniach radiologicznych),
- wpływ czynników zewnętrznych – brak dobrego oświetlenia lub też pomiar obiektu, który znajduje się w ruchu może znacznie utrudnić akwizycję obrazu,
- straty w trakcie transmisji przez kanał komunikacyjny – nieodpowiednio zabezpieczony przesył sygnału z urządzenia pomiarowego może być podatny na dodatkowe zakłócenia,
- błędy ludzkie – nawet jeśli pierwsze trzy czynniki mają minimalny wpływ na jakość obrazu, może się zdarzyć, że osoba która przygotowuje obrazy ustawi nieodpowiednie parametry w urządzeniu akwizycyjnym, lub nieodpowiednio przygotowuje obiekt badania i wyniki będą dalekie od oczekiwanych,
- ograniczenia finansowe – często decydujący czynnik, który wymusza korzystanie ze sprzętu nie najnowszej generacji i gorsze przygotowanie obiektu do badań, lub też pośpiech w trakcie ich wykonywania.

Te i inne powody wprowadzają do obrazu niepotrzebne zakłócenia i artefakty, które mogą utrudnić jego właściwą analizę. Człowiek w naturalny sposób patrząc na obrazy niskiej jakości jest w stanie skupić uwagę na kluczowych obiektach i opierając się na doświadczeniu ocenić, czy przedstawiona na obrazie sytuacja jest poprawna, czy też nie. Tego rodzaju proces filtracji jest niezbędny również w przypadku cyfrowego przetwarzania obrazu i powinien być stosowany przed uruchomieniem docelowych algorytmów segmentacji obrazu.



## 3.1 Proste filtry korzystające z sąsiedztwa punktu

Jedną z najprostszych metod usuwania szumu jest filtracja bazująca na lokalnym sąsiedztwie każdego punktu obrazu. Podejście to korzysta z uproszczonego założenia, że punkt i jego bezpośrednie otoczenie należy do tego samego obiektu i powinny one mieć zbliżone wartości lub wartości te powinny zmieniać się stopniowo. Przy takim założeniu wartości znacznie różniące się od pozostałych uznaje się za zakłócone przez czynniki zewnętrzne i powinny one zostać zmodyfikowane. Często stosowaną techniką w takim przypadku jest uśrednianie każdego punktu na podstawie jego otoczenia. Najczęściej stosowana wielkość otoczenia o promieniu jednego lub dwóch punktów pozwala zredukować drobny szum, natomiast znaczne zwiększenie tej wartości, np. do poziomu 5 lub 6 punktów, daje możliwość usunięcia drobnych elementów na obrazie, które nie są przedmiotem analizy i dodatkowo wygładza obiekty większe, które mogą być po takiej operacji łatwiejsze w detekcji. Zazwyczaj korzysta się dwóch rodzajów średniej:

- *Średnia arytmetyczna* – wygładza cały obraz ustawiając dla każdego punktu wartość średnią z jego otoczenia. Należy ostrożnie stosować to podejście ponieważ rozmywa ona granice między obiektami i wprowadza do obrazu nowe wartości.
- *Mediana* – nie wprowadza do obrazu nowych wartości, tylko wybiera dla modyfikowanego punktu wartość centralną z lokalnego otoczenia. Pozwala to na lepsze zachowanie granic, niż w przypadku średniej arytmetycznej, ale może wpływać na geometrię obrazu poprzez zaokrąglenie ostrych krawędzi lub usunięcie cienkich linii [60]. Mediana bywa szczególnie skuteczna do likwidacji szumu impulsowego.

Łatwo zauważyć, że te dwa podejścia działają według różnych przesłanek. W przypadku *średniej arytmetycznej* mamy do czynienia z podejściem liniowym, w którym niezależnie od położenia analizowanego punktu wykonywana jest identyczna operacja – uśrednianie. Natomiast w przypadku *mediany*, za każdym razem podejmowana jest indywidualna decyzja – wybór punktu centralnego. Jest to podejście nieliniowe. W kolejnych podrozdziałach zostaną przedstawione bardziej zaawansowane techniki zarówno liniowe, jak i nieliniowe, które częściowo rozwiązują wskazane problemy wyżej wymienionych metod.

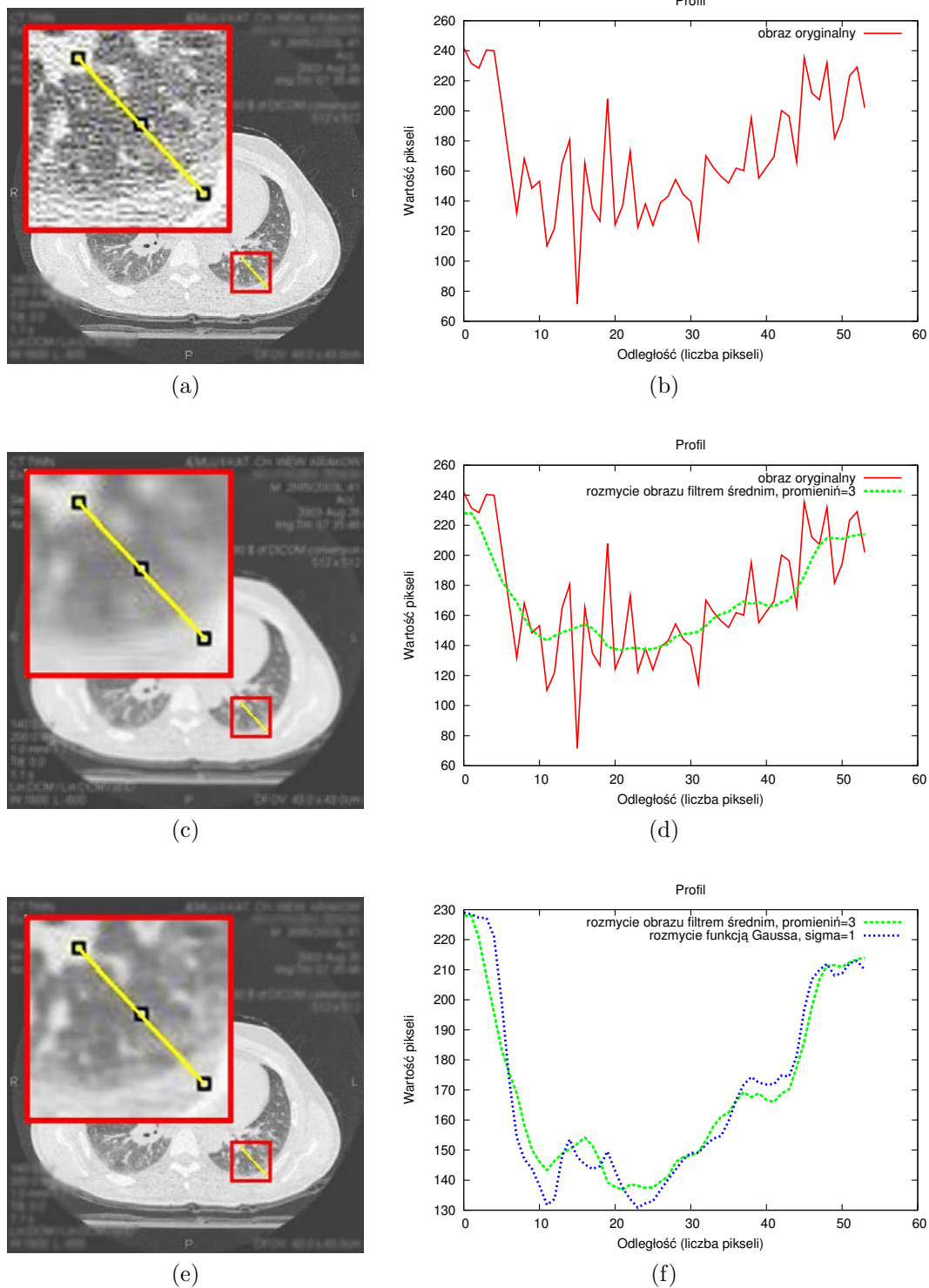
## 3.2 Liniowe wygładzenie obrazu bazujące na funkcji Gaussa

Stosowanie średniej arytmetycznej do usuwania szumu może prowadzić do nadmiernego rozmycia obrazu i utraty istotnych informacji o granicach obiektów. Wynika to w dużej mierze z faktu identycznego traktowania wszystkich punktów z przyjętego sąsiedztwa – zatem na punkt centralny taki sam wpływ mają punkty do niego przyległe, jak i te leżące na skraju przyjętego otoczenia. Własność ta może negatywnie wpływać na punkty leżące na styku dwóch obiektów, lub w pobliżu ich granic, co jest szczególnie widoczne dla większych wartości sąsiedztwa. Rozpatrując przykładowe sąsiedztwo o promieniu 3 punktów, do liczenia średniej uwzględniamy 49 elementów (obszar 7 na 7 punktów), gdzie każda wartość będzie miała taki sam wpływ na końcową wartość punktu centralnego. Można to przedstawić za pomocą macierzy filtru widocznej w tabeli 3.1.

Tabela 3.1: Macierz filtru uśredniającego o rozmiarach 7 x 7.

$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$
$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$
$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$
$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	<b>49</b>	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$
$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$
$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$
$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{49}$

Taka wielkość filtru bardzo dobrze wygładzi obszary znajdujące się wewnątrz obiektów, jednak doprowadzi do znacznej utraty ostrości ich granic (Rys. 3.1c). Aby chociaż częściowo zapobiec temu zjawisku często stosowaną techniką jest różnicowanie wpływu przyległych punktów w zależności od ich odległości od punktu centralnego. Powstała w ten sposób średnia ważona preferuje najbliższe punkty, których wartości powinny być najbardziej zbliżone do punktu centralnego.



Rysunek 3.1: Wyniki działania liniowych filtrów wygładzających dla przykładowego obrazu tomograficznego klatki piersiowej: (a) obraz oryginalny, (c) filtracja uśredniająca o *promieniu* = 3, (e) filtracja Gaussa o wartości *sigma* = 1. W prawej kolumnie przedstawiono profile wartości pikseli dla wybranej linii (*komentarz w tekście, opracowanie własne*).

Wygodnym sposobem ustalenia wartości poszczególnych wag jest zastosowanie dwuwymiarowej funkcji rozkładu Gaussa:

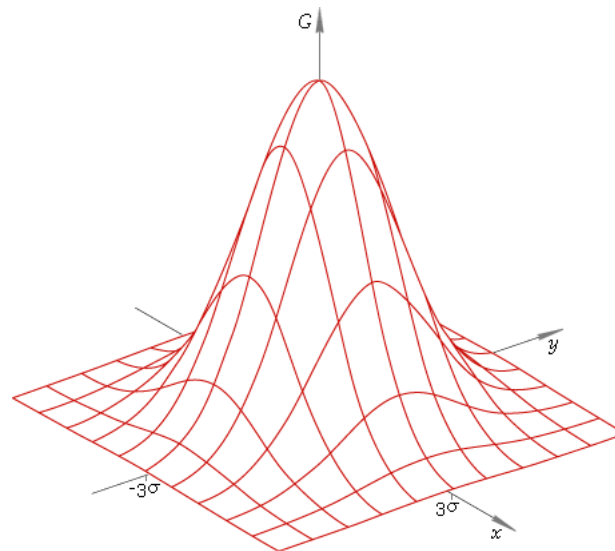
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.1)$$

gdzie:

$x$  – odległość od osi X,

$y$  – odległość od osi Y,

$\sigma$  – odchylenie standardowe rozkładu normalnego (*sigma*)<sup>1</sup>.



Rysunek 3.2: Wykres dwuwymiarowej funkcji rozkładu Gaussa (źródło: *Internet*).

Utworzony za pomocą tej funkcji wykres przypomina swym kształtem dzwon (Rys. 3.2). Tworząc docelową maskę filtru należy przypisać punktowi centralnemu największą wartość uzyskaną z wykresu, a dla pozostałych punktów obliczyć odpowiednie wartości korzystając z podanego wyżej wzoru. Aby ustalić rozmiar macierzy filtru warto zwrócić uwagę, że 99,7% pola pod wykresem znajduje się w odległości  $3\sigma$  od wartości środkowej. Dlatego w praktyce maski, które się konstruuje mają wymiary  $[6\sigma + 1] \times [6\sigma + 1]$ . Dodatkowo, po etapie dyskretyzacji funkcji Gaussa do macierzy filtru warto znormalizować tą macierz tak, aby mieć pewność, że suma wartości wszystkich jej elementów będzie się równać 1. Można to uzyskać dzieląc każdy element przez sumę wszystkich elementów. Obliczona wg tych zaleceń maska filtru dla wartości *sigma* = 1, będzie miała wielkość  $7 \times 7$  elementów i wartości takie, jak w tabeli 3.2.

<sup>1</sup>W tekście zamiennie jest stosowane oznaczenie symbolu odchylenia standardowego ( $\sigma$ ) z pełną nazwą (*sigma*).

Tabela 3.2: Macierz filtru Gaussa o wartości  $\sigma = 1$ .

0.00000067	0.00002292	0.00019117	0.00038771	0.00019117	0.00002292	0.00000067
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	0.00019117
0.00038771	0.01330373	0.11098164	<b>0.22508352</b>	0.11098164	0.01330373	0.00038771
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	0.00019117
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00000067	0.00002292	0.00019117	0.00038771	0.00019117	0.00002292	0.00000067

Wyniki działania omówionych liniowych filtrów wygładzających przedstawiono na rysunku 3.1. W celu lepszego zobrazowania uzyskanych rezultatów pokazano profile wartości dla wybranej linii (prawa kolumna) – zarówno dla obrazu oryginalnego, jak i dla obrazów po zakończonej filtracji. Dodatkowo obszar z zaznaczoną linią profilu został znacznie powiększony, aby lepiej pokazać zmiany wprowadzane na obrazie przez poszczególne filtry. Obserwując profil wartości pikseli dla obrazu źródłowego (Rys. 3.1a) można zauważyć ich wysoki rozrzut wewnątrz obszaru należącego do płuca (Rys. 3.1b). Zastosowanie filtracji uśredniającej, pomimo bardzo dobrego wygładzenia tych wartości (Rys. 3.1c i profil 3.1d), powoduje niestety nadmierne rozmycie granic obiektów. Lepiej w tym aspekcie radzi sobie filtr Gaussa, który nie wpływa tak bardzo na strukturę obiektów (Rys. 3.1e). Widać to szczególnie na wykresie 3.1f, gdzie zestawiono obok siebie profile działania obu filtrów. Na początku i na końcu tego wykresu, w miejscach gdzie obszar płuc łączy się z korpusem ciała, filtr Gaussa ma ostrzejsze zbocze, aniżeli filtr uśredniający. Oznacza to, że przejście jednego obszaru w drugi odbywa się na mniejszej odległości, co skutkuje tym, że granica między nimi jest ostrzejsza.

### 3.3 Nieliniowe wygładzenie obrazu bazujące na funkcji Gaussa

Pomimo tego, że rozmycie funkcją Gaussa lepiej sobie radzi z zachowaniem granic między obiektami niż zastosowanie średniej arytmetycznej, to w rzeczywistości każdy punkt jest tak samo przetwarzany, niezależnie od tego czy leży wewnątrz obiektu, czy też w pobliżu jego granicy. Jest to znana cecha filtrów liniowych. Skuteczniejszym rozwiązaniem mogłoby się okazać indywidualne traktowanie każdego punktu w zależności od jego lokalizacji, tak aby proces filtracji był wzmocniony na obszarach jednolitych, a zahamowany w pobliżu krawędzi.

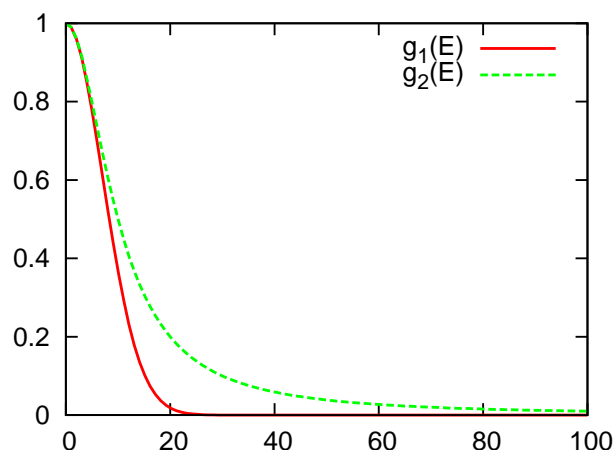
Jedną z pierwszych prac w tym zakresie był artykuł Perona i Malika [53]. Opisana w nim metoda nosi od nazwisk autorów nazwę *modelu Perona-Malik*<sup>2</sup>. Proponuje się w nim wprowadzenie zmiennego współczynnika  $g$  decydującego o stopniu rozmycia w zależności od wartości estymatora krawędzi  $E$ . Jako przykład estymatora krawędzi autorzy proponują gradient jasności obrazu w punkcie zdefiniowany jako:

$$E(x, y) = \|\nabla u(x, y)\| \quad (3.2)$$

W cytowanej pracy podane zostały również dwie postacie funkcji  $g$ , wpływającej na stopień rozmycia obrazu (wykresy funkcji pokazane na rysunku 3.3):

$$g_1(E) = \exp\left(-\left(\frac{E}{K}\right)^2\right) \quad (3.3)$$

$$g_2(E) = \frac{1}{1 + \left(\frac{E}{K}\right)^2} \quad (3.4)$$

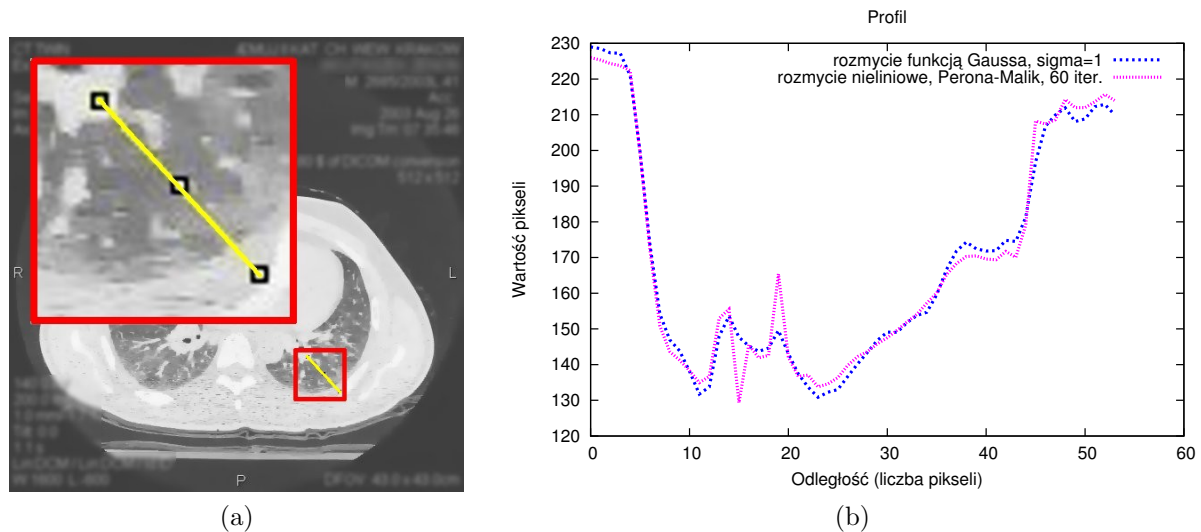


Rysunek 3.3: Wykres zależności współczynników wygładzenia obrazu  $g_1$  i  $g_2$  od wartości estymatora krawędzi  $E$  (*opracowanie własne*).

Dodatkowy parametr  $K > 0$  wskazuje na minimalną wartość potrzebną do zakwalifikowania wybranego punktu do krawędzi bądź też jego odrzucenia. Może być on rozumiany jako różnica jasności pomiędzy krawędzią, a otoczeniem. Parametr ten może być eksperymentalnie wyznaczany przez użytkownika lub też może być zależny od histogramu szumu wyznaczonego z obrazu [53].

<sup>2</sup>W artykule została przedstawiona matematyczna interpretacja algorytmu z użyciem procesu dyfuzji ciepła, która tutaj zostanie pominięta jako wykraczająca poza ramy niniejszej pracy. Bardziej szczegółowy opis tej oraz innych metod nieliniowych filtracji obrazu można znaleźć w pracy [30].

W odróżnieniu od wcześniej opisywanych filtrów liniowych, *model Perona-Malik* operuje tylko na punktach z bezpośredniego otoczenia (promień filtru = 1). Aby umożliwić regulację stopnia rozmycia zaleca się wielokrotne filtrowanie obrazu<sup>3</sup> (algorytm zalicza się do algorytmów iteracyjnych). Przykład działania omawianego filtru nieliniowego



Rysunek 3.4: Wyniki działania nieliniowego filtru rozmycia obrazu opartego o *model Perona-Malik*: (a) obraz po filtracji (60 iteracji), (b) profil wartości dla wybranej linii porównany z profilem obrazu po rozmyciu funkcją Gaussa (*opracowanie własne*).

przedstawiono na rysunku 3.4. Tutaj również powiększono wybrany fragment obrazu i utworzono wykres z profilem wartości. Profil dodatkowo porównano z profilem z obrazu po filtracji Gaussa (Rys. 3.1f). Profile te pomimo tego, że w przeważającej części wydają się podobne, wykazują pewne różnice w sąsiedztwie krawędzi obiektów. Profil dla obrazu po filtracji nieliniowej charakteryzuje się w tych miejscach wyższymi skokami wartości pikseli, aniżeli profil obrazu po filtracji funkcją Gaussa. Obserwując również wizualnie wyniki filtracji na obrazie 3.4a widać, że granice poszczególnych obiektów nie zostały rozmyte, a dzięki temu, że został usunięty szum z obrazu sprawiają one nawet wrażenie ostrzejszych niż na obrazie źródłowym 3.1a.

<sup>3</sup>Jest to bezpośrednio związane z naturą procesu dyfuzji ciepła, na której autorzy opierali się podczas tworzenia algorytmu, a której efekty zależą od czasu.

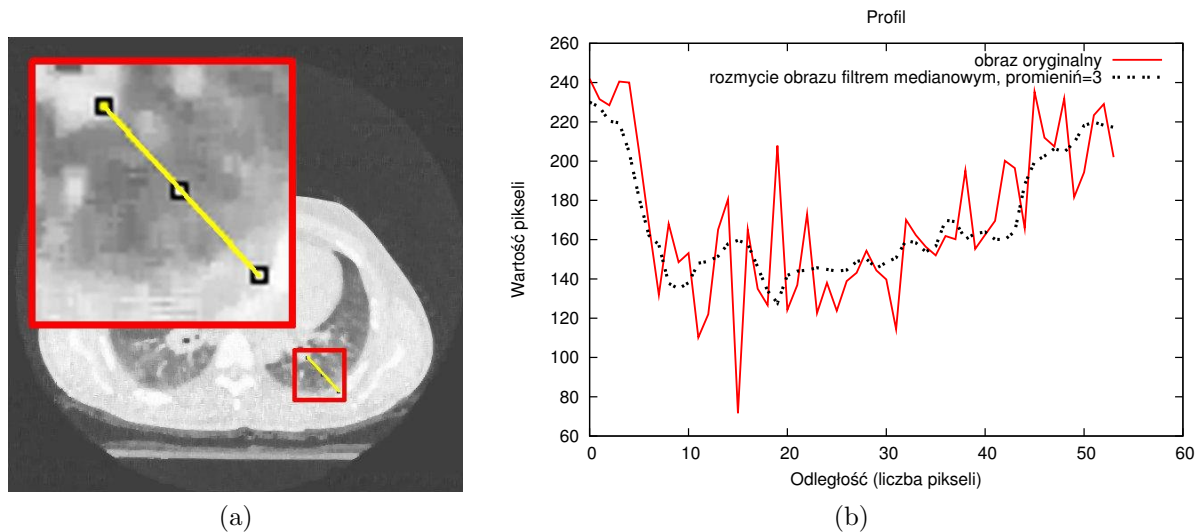


W przedstawionym modelu nieliniowej filtracji obrazu bazującej na *modelu Perona-Malik* słabym ogniwem może okazać się sposób obliczania estymatora krawędzi. Przykładowa metoda bazująca na lokalnym gradiencie jest bardzo czuła na szum i w miejscach gdzie jego poziom jest bardzo wysoki może się okazać, że punkt zostanie zakwalifikowany do krawędzi, zamiast do obszaru jednorodnego i te obszary nie zostaną wygładzone. Aby temu zaradzić w pracy [8] proponuje się wprowadzenie dodatkowego **etapu regularyzacji modelu**. Może się to odbywać z wykorzystaniem liniowych funkcji rozmycia obrazu (zalecane rozmycie Gaussa), które wygładzą nadmierne skoki wartości punktów. Dopiero na tak przygotowanym obrazie wyliczenie estymatora krawędzi nie będzie zależne od lokalnych zakłóceń i umożliwi bardziej wiarygodne wygładzenie obrazu właściwą *metodą Perona-Malik*. Warto również zwrócić uwagę na to, że regularyzacja modelu wykonywana jest tylko przed pierwszym wyliczeniem estymatora krawędzi i w kolejnych iteracjach nie musi być już stosowana, aby nie prowadzić do niepotrzebnej utraty informacji o położeniu krawędzi.

Opisana powyżej filtracja nieliniowa pomimo tego, że daje lepsze wyniki aniżeli tradycyjny filtr uśredniający oraz klasyczny filtr Gaussa, nie jest wolna od wad. Głównym problemem w jej stosowaniu jest przede wszystkim długi czas działania, wynikający z iteracyjnej natury tego typu algorytmów. Dla przykładu, aby uzyskać rezultat podobny do tego przedstawionego na rysunku 3.4a, należało wykonać 60 iteracji algorytmu dla całego obrazu. Jest to nieporównywalnie więcej od tradycyjnych filtrów takich jak średnia, lub mediana, które potrzebują wyłącznie raz ustalić wartość każdego punktu obrazu. Dodatkowym problemem w stosowaniu omawianych filtrów nieliniowych może być odpowiednie dopasowanie liczby iteracji do otrzymania dobrego rezultatu. Pomimo tego, że algorytm ten nie jest podatny na niewielkie zmiany liczby iteracji, to i tak dla osoby, która nie ma dużego doświadczenia w posługiwaniu się tego typu metodami może nie być jasne, czy liczba iteracji powinna wynosić 60, czy też 120, czy z drugiej strony powinna zostać zmniejszona do 30. Inny problem, o którym warto tutaj wspomnieć, jest mniejsza popularność tego typu algorytmów, przez co może się okazać, że nie znajdziemy ich w wykorzystywanych przez nas do tej pory narzędziach. Natomiast próba ich samodzielnej implementacji może być kłopotliwa, ponieważ są one bardziej skomplikowane algorytmicznie, aniżeli tradycyjne filtry liniowe.



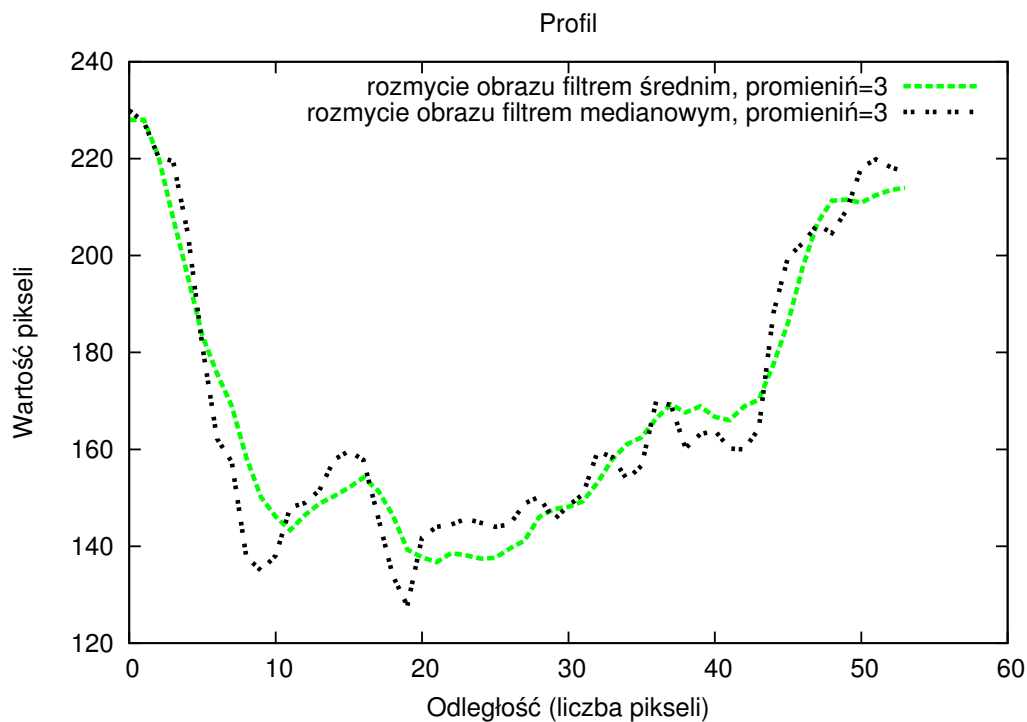
Jeśli tego typu wady (długi czas działania, kwestia doboru liczby iteracji i problemy w implementacji) okażą się problematyczne, to rozwiązaniem może okazać się zastosowanie innego algorytmu nieliniowego, o którym była mowa na początku tego rozdziału, czyli mediany. Wybierając ten algorytm, który też nie jest bez wad (napisano o tym na początku



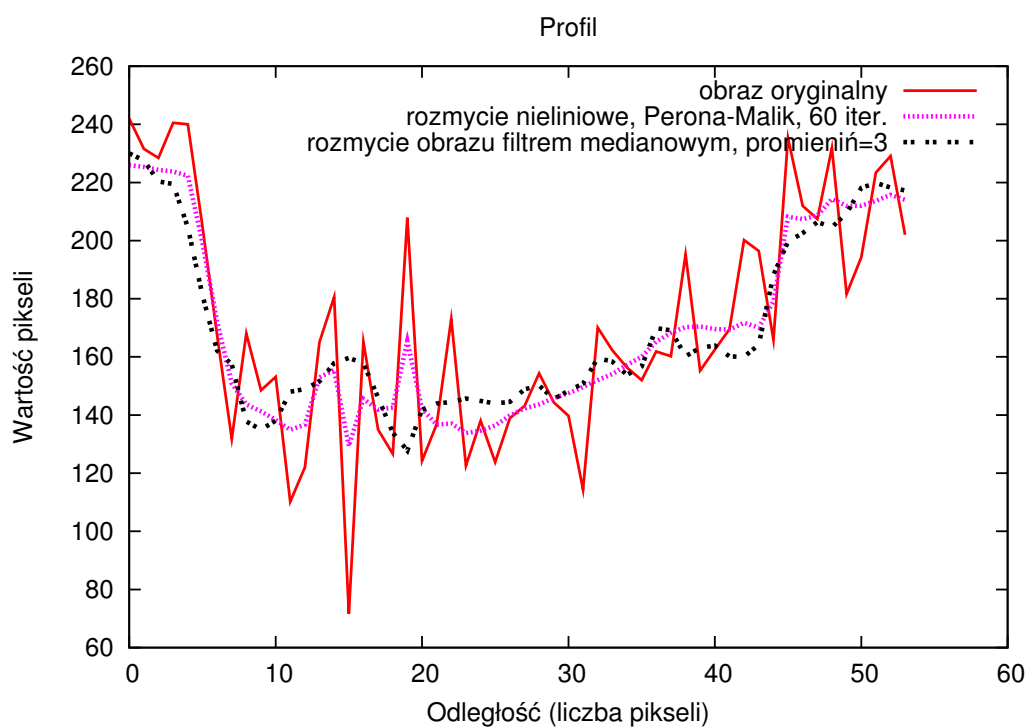
Rysunek 3.5: Wyniki działania filtru medianowego: (a) obraz po filtracji, (b) profil wartości dla wybranej linii (*opracowanie własne*).

tego rozdziału) może okazać się, że otrzymamy w zupełnie satysfakcjonujące rezultaty. Wystarczy tylko spojrzeć na rysunek 3.5a, aby upewnić się, że po zastosowaniu tej filtracji granice obiektów ciągle pozostają ostre. Potwierdza to również znajdujący się obok na rysunku 3.5b profil przebiegający po tych samych punktach co poprzednio (Rys. 3.1a, 3.1c, 3.1e i 3.4a). W odróżnieniu od filtracji średniej (porównanie profili mediany i średniej przedstawiono na rysunku 3.6) usunięcie skrajnych wartości nie powoduje nadmiernego rozmycia obrazu i pozwala na zachowanie ostrości brzegu poszczególnych obiektów. Natomiast zestawiając profile wyników filtracji medianowej i filtracji *Perona-Malik* z obrazem oryginalnym (porównanie profili zamieszczono na rysunku 3.7) trudno wydać jednoznaczny werdykt, który z tych algorytmów jest lepszy<sup>4</sup>. Pomimo tego, że oba filtry dobrze radzą sobie w miejscach pojedynczych skoków wartości pikseli, to wydaje się, że profil filtru *Perona-Malik* lepiej pokrywa się ze z profilem obrazu oryginalnego w miejscach, gdzie następuje rzeczywista zmiana jednego obiektu w drugi.

<sup>4</sup>Dokładne porównanie tych dwóch filtracji obrazu wymaga opracowania odpowiednich miar jakości wygładzenia obrazu, przy jak najlepszym zachowaniu ostrości granic, a następnie wykonaniu szeregu porównań dla obrazów o różnym poziomie zaszumienia. Tego typu badania, mimo iż są bardzo ważne z punktu widzenia komputerowej analizy obrazu, wykraczają poza ramy niniejszej pracy.



Rysunek 3.6: Porównanie profilu filtracji średnią arytmetyczną (Rys. 3.1c) z medianą (Rys. 3.5a) (opracowanie własne).



Rysunek 3.7: Porównanie profilu filtracji medianowej (Rys. 3.5a) i filtracji *Perona-Malik* (Rys. 3.4a) z obrazem oryginalnym (Rys. 3.1a) (opracowanie własne).

### *Wstępna segmentacja obrazu*

W rozdziale 2 poświęconym analizie aktualnego stanu prac w zakresie tematyki niniejszej pracy pokrótce omówiono algorytmy segmentacji obrazu i możliwości ich użycia dla obrazów niejednorodnych. Wskazano tam na problem określenia kryterium jednorodności pikseli, które wykorzystywane jest do grupowania pikseli w obszary odpowiadające rzeczywistym obiektom na obrazie. Pomimo tego, że ciągle konstruowane są coraz to nowe i bardziej zaawansowane algorytmy segmentacji obrazu, trudno uznać, aby w równie dobrym stopniu sprawdzały się one w każdym podejmowanym zastosowaniu. Dlatego jeśli analiza dotyczy dużych serii obrazów przedstawiających podobną scenę, najlepszym rozwiązaniem może być konstruowanie dedykowanych algorytmów, w których szczegółowo opisuje się cechy charakterystyczne kluczowych obiektów (np. informacje o teksturze, kolorze, krawędziach). Algorytmy tego typu, pomimo wprowadzenia pewnych ograniczeń w zakresie ich stosowania, często okazują się bardziej efektywne i dają bardziej precyzyjne wyniki od klasycznych algorytmów segmentacji obrazu.

Postawiona w niniejszej pracy teza zakłada możliwość skonstruowania metod, które, nie tracąc ogólnej natury, mogą być skutecznie używane do segmentacji i detekcji dużych obiektów. Kluczowe w tym celu będzie odpowiednie zamodelowanie informacji o tych obiektach i przekazanie jej do właściwego algorytmu<sup>1</sup>. Metody te w dużym stopniu będą opierać się na istniejących algorytmach segmentacji obrazu, które dobrze radzą sobie z wydzieleniem elementarnych struktur obrazu - mniejszych obiektów, które wchodzi w skład dużych obiektów. Przygotowana w ten sposób wstępna segmentacja obrazu zostanie użyta do tworzenia dużych obiektów z wielu obiektów mniejszych. W dalszej części rozdziału zostanie opisany algorytm działów wodnych oraz jeden z algorytmów segmentacji obrazu bazujący na teorii grafów.

---

<sup>1</sup>Odpowiednie algorytmy i sposoby przekazywania informacji zostaną zaproponowane w III części pracy.

## 4.1 Klasyczny algorytm działów wodnych

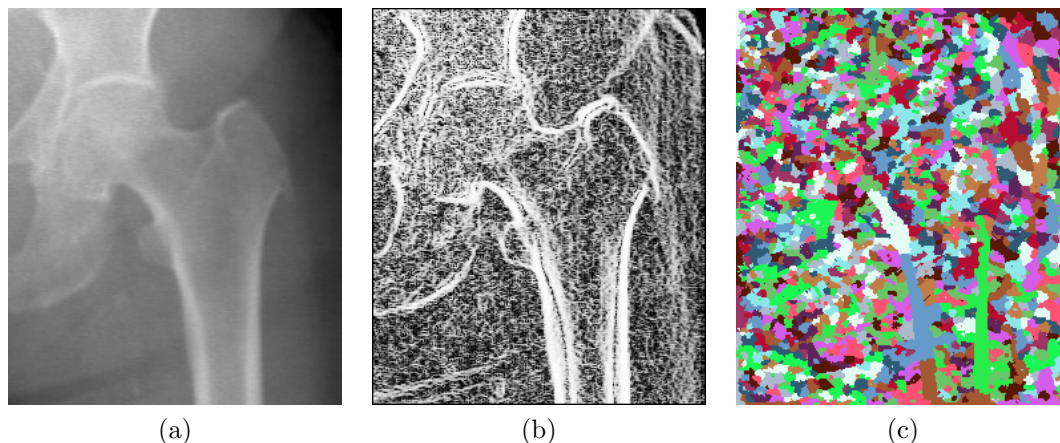
Aby zobrazować algorytm działów wodnych [73] jego działanie zostanie porównane do zalewania terenu przez wodę. W tym celu warto spojrzeć na obraz niczym na mapę topograficzną terenu. Analogia będzie jak najbardziej prawdziwa, jeśli założymy, że jasność poszczególnych pikseli określa stopień wzniesienia względem określonego punktu odniesienia – w przypadku mapy jest nim poziom morza. Dodatkowo, na potrzeby ilustracji omawianego algorytmu można założyć, że podłoże terenu ma strukturę gąbki i nie ma problemu z przepuszczaniem wilgoci. Przy tak postawionych założeniach można wyobrazić sobie proces równomiernego zalewania (nasiąkania) całego terenu od dołu do góry. Oczywistym wydaje się, że najpierw zamokną tereny położone najniżej, a woda będzie równomiernie się w nich wypełniać, aż połączy się z sąsiednimi zlewiskami.

Niestety stosowanie tego podejścia do obrazów w oryginalnej postaci może dać niepożądane rezultaty. Powodem tego jest fakt, że obiekty na obrazie mogą mieć różną jasność względem siebie i względem tła. W ten sposób najpierw zostałyby wypełnione najciemniejsze obiekty, później tło, a na samym końcu obiekty jaśniejsze od tła. Tym samym część obiektów (tych najciemniejszych) mogłaby być pominięta. Jeszcze większy problem może przysporzyć analiza obiektów niejednorodnych, których jasność jest inna w różnych jego częściach. Biorąc pod uwagę wymienione problemy zaleca się przygotowanie najpierw obrazu gradientowego<sup>2</sup>, który uwydatnia krawędzie obiektów i dopiero później stosowanie algorytmu działów wodnych. W obrazie gradientowym *lokalne minima*, czyli punkty, które jako pierwsze zalewane są wodą w poszczególnych segmentach<sup>3</sup>, leżą w strefach jednolitego nasycenia kolorów, natomiast czym bliżej krawędzi obiektów tym wartości gradientu są wyższe i to one będą stanowić punkty łączenia się dwóch segmentów. Wprowadzając nazewnictwo związane z analizą obrazu można powiedzieć, że punkty łączenia się zlewisk będą stanowić *linię wododziału*, natomiast same zlewiska będą określać *niezależne segmenty obrazu*.

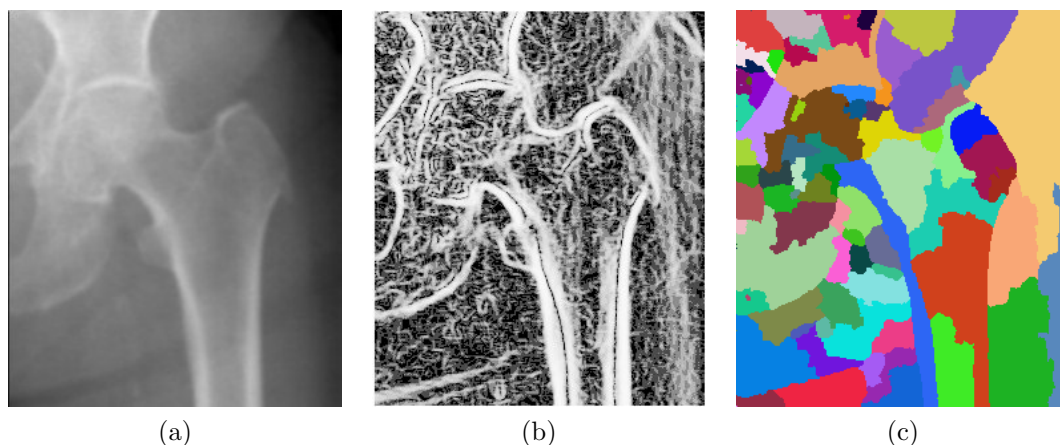
Wyniki działania algorytmu działów wodnych przebiegającego wg opisanej koncepcji można zobaczyć na rysunku 4.1. Ilość segmentów, które powstały w wyniku działania algorytmu zdecydowanie przewyższa liczbę obiektów, które mógłby wskazać użytkownik. W tym wypadku problem jest związany z występowaniem szumu na całym obrazie, który wprowadza znaczny nadmiar lokalnych minimów. W takiej sytuacji dobrym podejściem będzie skorzystanie z algorytmu usuwania szumu i ponowne uruchomienie całej procedury od początku. Rezultaty pokazane na rysunku 4.2 są znacznie lepsze, niż poprzednie. Do rozmycia obrazu użyto filtracji Gaussa o parametrze  $\sigma = 1$ . Nawet jeśli odwzorowanie

<sup>2</sup>Stosowanie operatorów gradientowych opisano w dodatku A.1.Proste sposoby detekcji krawędzi.

<sup>3</sup>Lokalne minima są to punkty, względem których wszyscy sąsiedzi mają większe wartości.



Rysunek 4.1: Wyniki segmentacji kości biodrowej z wykorzystaniem algorytmu działów wodnych [73]: (a) obraz oryginalny, (b) obraz gradientowy, (c) wynik segmentacji (*opracowanie własne*).



Rysunek 4.2: Segmentacja wododziałowa obrazu po uprzednim usunięciu szumu: (a) oryginalny obraz po rozmyciu filtrem Gaussa o wartości  $\sigma = 1$ , (b) obraz gradientowy, (c) wyniki segmentacji (*opracowanie własne*).

rzeczywistych obiektów na segmenty ciągle nie jest poprawne, to uzyskany wynik może być dobrym punktem wyjścia do dalszej analizy.

Efekt działania opisanego algorytmu w istotny sposób zależy od liczby i położenia punktów startowych dla procesu zalewania. W klasycznym algorytmie – tak jak przedstawiono powyżej – korzysta się z lokalnych minimów, a wpływać na nie można stosując wygładzanie obrazu. Możliwe jest oczywiście przygotowanie dedykowanego algorytmu wyznaczającego punkty startowe lub poproszenie użytkownika o ich manualne wskazanie. To drugie podejście zostanie szerzej omówione w rozdziale 8, jako jedna z proponowanych metod przygotowania modelu opisującego duże obiekty.

## 4.2 Segmentacja z wykorzystaniem minimalnego drzewa rozpinającego

Zastosowanie reprezentacji grafowej na potrzeby analizy obrazu pozwala skorzystać z licznych algorytmów operujących na grafach, m.in. algorytmów poszukiwania optymalnych ścieżek, drzew rozpinających i cięć grafów<sup>4</sup>. Algorytmy te rozpoczynają swoje działanie od lokalnych obszarów i w trakcie swojego działania pozyskują coraz więcej informacji o otoczeniu. Dzięki temu w procesie analizy zostają uwzględnione zarówno lokalne jak i globalne cechy poszczególnych obiektów.

W rozdziale tym zostaną opisane możliwości wykorzystania algorytmu wyznaczającego minimalne drzewo rozpinające (ang. *Minimum spanning trees, MST*) w procesie segmentacji obrazu, które zostały przedstawione w pracach [76] i [18]. Na potrzeby tego algorytmu, segmentacja obrazu będzie rozważana jako podział zbioru pikseli na wzajemnie rozłączne podzbiory  $C_1, C_2, C_3, \dots, C_N$ , gdzie każdy piksel należy do jednego z podzbiorów  $C_i$ . Dokonując transformacji obrazu na graf, każdy piksel będzie reprezentowany jako wierzchołek tego grafu, natomiast krawędzie łączące sąsiadujące piksele przyjmą wagi równe różnicy ich wartości. W celu przeprowadzenia dalszej analizy algorytmu zostaną wprowadzone następujące oznaczenia [18]:

- wewnętrzne zróżnicowanie podzbioru  $C_i$ :

$$Int(C_i) = \max w(e), \quad e \in MST(C_i) \quad (4.1)$$

gdzie:

$MST(C_i)$  – minimalne drzewo rozpinające podzbioru  $C_i$ ,

$e$  – krawędź należąca do  $MST$ ,

$w(e)$  – waga krawędzi  $e$ ,

$Int(C_i) = 0$  – dla zbiorów jednoelementowych.

- różnica między dwoma podzbiarami  $C_m$  i  $C_n$ :

$$Diff(C_m, C_n) = \min w(V_i, V_j), \quad V_i \in C_m, V_j \in C_n, \{V_i, V_j\} \in E, \quad (4.2)$$

gdzie:

$V_i, V_j$  – wierzchołki należące odpowiednio do  $C_m$  i  $C_n$ ,

$w(V_i, V_j)$  – waga krawędzi łączącej  $V_i$  i  $V_j$  (o ile taka krawędź istnieje).

---

<sup>4</sup>Zwięzłe wprowadzenie do możliwości zastosowania algorytmów grafowych na potrzeby analizy obrazu można znaleźć w pracy [68].



- minimalna różnica wewnętrzna podzbiorów  $C_m$  i  $C_n$ <sup>5</sup>:

$$MInt(C_m, C_n) = \min\left\{Int(C_m) + \frac{\tau}{|C_m|}, Int(C_n) + \frac{\tau}{|C_n|}\right\} \quad (4.3)$$

gdzie:

$\tau$  – parametr zdefiniowany przez użytkownika,

$| \cdot |$  – oznacza liczbę elementów zbioru.

Opierając się na wyżej postawionych założeniach, dwa segmenty  $C_m$  i  $C_n$  należą do tego samego obiektu, jeśli wartość  $Diff(C_m, C_n)$  jest mniejsza od  $MInt(C_m, C_n)$ . Działanie tak określonego algorytmu rozpoczyna się od segmentacji, w której każdy piksel tworzy osobny podzbiór  $C_i$  i rosnącego przesortowania wag krawędzi. Następnie rozpoczynając od krawędzi o najmniejszej wadze, sprawdza się, korzystając z wyżej określonych warunków, czy krawędź ta może połączyć dwa podzbiory. W każdym kroku otrzymuje się uaktualniony podział zbioru pikseli na podzbiory  $C_1, C_2, C_3, \dots, C_i, \dots$ .

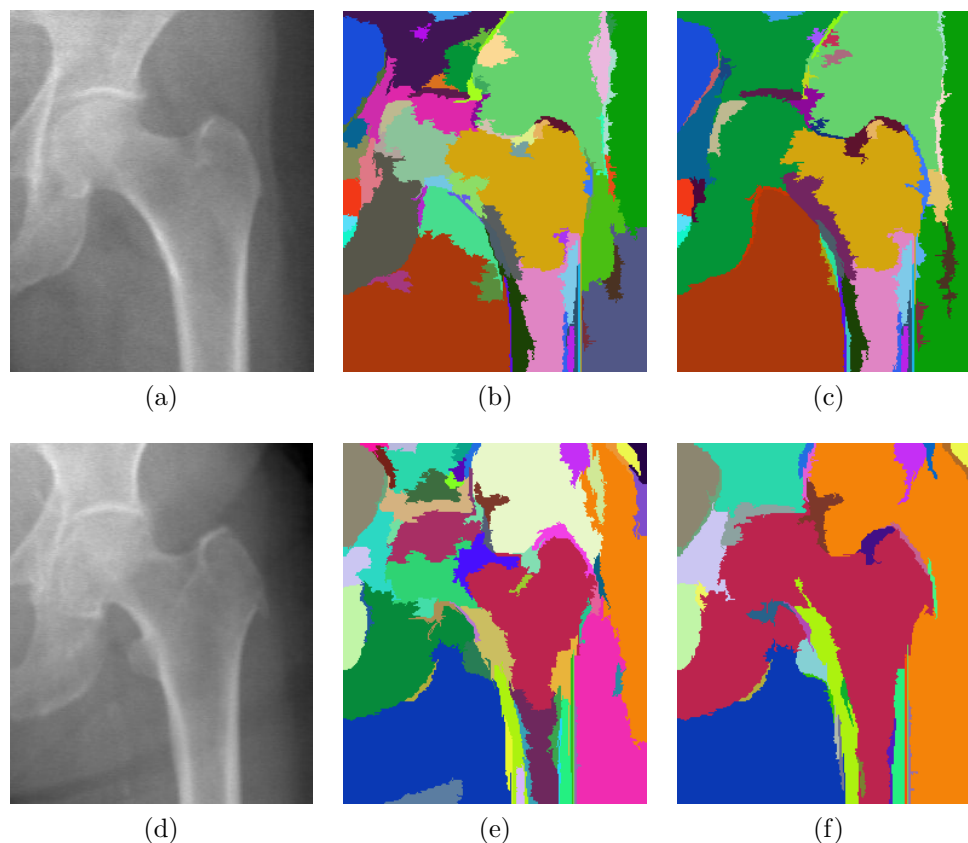


Rysunek 4.3: Przykład segmentacji obrazu niejednorodnego algorytmem grafowym poszukującym minimalnego drzewa rozpinającego:

(a) obraz wejściowy, (b) wynik segmentacji (*opracowano na podstawie [18]*).

W artykule [18] autorzy algorytmu pokazują możliwości wykorzystania omówionej koncepcji na przykładzie sztucznie wygenerowanych obrazów niejednorodnych (Rys. 4.3). Wskazują oni na możliwości poprawnej segmentacji zarówno obiektów o różnej teksturze, jak i tych, w których występują delikatne przejścia tonalne. Własności te niekoniecznie są pożądane w przypadku analizy rzeczywistych obrazów niejednorodnych, ponieważ mogą wpływać na łączenie sąsiednich obiektów pomiędzy, którymi nie ma ostrej granicy. Testy przeprowadzone z wykorzystaniem przykładowych zdjęć rentgenowskich kości udowej (Rys. 4.4) dla niskiej wartości progowej  $\tau = 40$  pozwoliły na uzyskanie zbliżonej wielkości segmentów (Rys. 4.4b i 4.4e), jak miało to miejsce w przypadku stosowania

<sup>5</sup>Oryginalny ang. termin: *minimum internal difference*.



Rysunek 4.4: Segmentacja rentgenowskich obrazów kości udowej (lewa kolumna) algorytmem grafowym *MST* [18] o wartości granicznej  $\tau = 40$  (środkowa kolumna) oraz  $\tau = 70$  (prawa kolumna) (*opracowanie własne*).

algorytmu działów wodnych po rozmyciu obrazu filtrem Gaussa o parametrze  $\sigma = 1$  (Rys. 4.2). Zwiększenie tego parametru do wartości  $\tau = 70$  w celu uzyskania większych segmentów powoduje scalenie głowy kości udowej z fragmentem obszaru należącego do miednicy (Rys. 4.4c i 4.4f). Kolejne zwiększanie wartości progowej  $\tau$  na większości testowanych obrazów prowadziło do szybszego łączenia obszarów należących do kości z tłem, zamiast łączenia się segmentów właściwych obiektów. Podobna sytuacja miała również miejsce w przypadku algorytmu działów wodnych, dlatego oba algorytmy mogą znaleźć zastosowanie raczej do segmentacji podstawowych struktur na obrazie, niż do wydzielenia dużych obiektów.



### *Łączenie segmentów*

Z uwagi na niebezpieczeństwo tworzenia niepoprawnych segmentów, które zawierają fragmenty kilku obiektów, w opisywanych w rozdziale 4 algorytmach zaleca się ostrożne zwiększanie parametrów wpływających na wielkość segmentów. Bezpieczniej jest uzyskać nadmiarowy podział obiektów na segmenty, niż doprowadzić do błędnych połączeń, tworząc w ten sposób segmenty składające się z fragmentów dwóch, lub więcej rzeczywistych obiektów. Opisane algorytmy segmentacji powinny być stosowane na wczesnym etapie analizy obrazu do wydzielenia podstawowych struktur. W kolejnym kroku przy użyciu dodatkowej informacji o własnościach tych struktur, będą mogły zostać one połączone we właściwe obiekty. Algorytmy za to odpowiadające noszą miano algorytmów łączenia (scalania) segmentów (ang. *Region-Merging Segmentation Algorithms*). Wymagają one określenia pewnych miar spójności, które powinny spełniać sąsiadujące segmenty w celu ich połączenia w obiekty odpowiadające rzeczywistym obiektom na obrazie.

W dalszej części tego rozdziału omówiono ideę działania algorytmów łączenia segmentów, a także przedstawiono popularne kryteria spójności. Celem tej analizy jest dokładniejsze wskazanie problemów, jakie stwarza użycie tego typu metod do analizy dużych obiektów na obrazach niejednorodnych. Dodatkowo, bliższe przedstawienie ich zasady działania umożliwi łatwiejszą analizę zaproponowanego w rozdziale 6 nowego algorytmu scalania regionów.

### **5.1 Wykorzystanie reprezentacji grafowej na potrzeby algorytmu łączenia segmentów**

Wygodnym sposobem konstruowania i implementacji algorytmów łączenia segmentów jest wykorzystanie reprezentacji grafowej. Zazwyczaj informacja o poszczególnych segmentach zapisywana jest w wierzchołkach grafu, a zależności między sąsiednimi segmentami zapamiętane są w jego krawędziach. Takie podejście znacznie ułatwia analizę zależności między poszczególnymi segmentami i podejmowanie decyzji o ich połączeniu.

Ogólna zasada działania algorytmów łączenia segmentów w oparciu o reprezentację grafową nie jest skomplikowana i może przebiegać według następującego schematu:

**krok 1** *Indeksacja* wyników segmentacji obrazu, która ma na celu przypisanie unikatowych wartości pikselom należącym do różnych segmentów.

**krok 2** Przygotowanie grafu reprezentowanego przez listy sąsiedztwa - dla każdego segmentu tworzona jest lista, w której zapisywane są indeksy przyległych segmentów.

**krok 3** Dla każdego segmentu wyliczana jest wartość spójności jego punktów według przyjętej miary (popularne miary opisano w kolejnej sekcji tego rozdziału), która następnie zapisywana jest jako waga wierzchołka grafu, odpowiadającemu temu segmentowi.

**krok 4** Dla każdej pary segmentów wyliczana jest wartość kryterium łączenia (najczęściej liczona jako różnica wartości miar), która następnie zapisywana jest jako waga krawędzi łączącej te segmenty.

**krok 5** Następnie wybierana jest krawędź o najniższej wadze i sprawdzane jest czy mieści się ona w dopuszczalnej granicy<sup>1</sup>

- Jeśli tak: segmenty połączone tą krawędzią scalane są w jeden segment - wykonywane są wtedy następujące operacje:

**krok 5.1** wszystkie punkty obu segmentów otrzymują ten sam indeks,

**krok 5.2** na podstawie poprzednich list tworzona jest nowa lista sąsiedztwa,

**krok 5.3** informacja o połączeniu przekazywana jest do wszystkich segmentów przyległych, tak aby mogły one również dokonać aktualizacji (zamienić stary indeks na nowy, lub usunąć jeden z indeksów, jeśli graniczyły z obydwojma połączonymi segmentami),

**krok 5.4** wyliczana jest ponownie miara spójności dla nowo powstałego segmentu oraz wartości kryterium łączenia z sąsiednimi segmentami; skutkuje to uaktualnieniem wag krawędzi grafu, z którymi łączy się ten segment,

**krok 5.5** Krok (5) wykonywany jest ponownie.

- Jeśli nie: zakończenie algorytmu.

Analizując ten algorytm łatwo zwrócić uwagę na jego iteracyjny charakter. Po etapie analizy wszystkich segmentów, algorytm wybiera te pary segmentów, które mają najniższą wartość kryterium łączenia. Dzięki temu następuje systematyczny wzrost najbardziej jednorodnych obszarów, który jest kontrolowany przez wartość graniczną podaną przez użytkownika.

---

<sup>1</sup>Wartość graniczna łączenia segmentów przekazywana jest jako parametr do algorytmu.

## 5.2 Popularne kryteria łączenia segmentów

Istnieje wiele miar służących do opisywania spójności pikseli w obrębie segmentów na obrazie. Pod pojęciem miary spójności można rozumieć pewną wielkość statystyczną (np. średnią, wariancję, korelację i inne), która pozwala na zmierzenie określonej cechy każdego segmentu (np. intensywności pikseli, własności histogramu). Później, na podstawie obliczonych wartości porównywane są sąsiadujące ze sobą segmenty i te z nich, które mają najbardziej zbliżone miary spójności, są ze sobą scalane. Czasami zamiast obliczania miar spójności dla każdego segmentu z osobna i porównywania ich ze sobą, warto dokładniej sprawdzić cechy wspólne sąsiadujących segmentów, np. poprzez analizę punktów wspólnych leżących na granicy między nimi. Dlatego w kontekście algorytmów łączenia segmentów, zamiast operowania na samych miarach spójności, często korzysta się z uogólnionego pojęcia tzw. **kryterium łączenia segmentów**, które dotyczy również innych pomiarów, niekoniecznie związanych z porównywaniem wspomnianych miar. Sterując wartością graniczną tego kryterium można wpływać na wielkość wynikowych segmentów.

Poniżej zostaną przedstawione często stosowane kryteria łączenia segmentów. W tym celu wprowadza się następujące oznaczenia:

- $K_i(S_1, S_2)$  - kryterium łączenia (podobieństwa) segmentów  $S_1$  i  $S_2$ ,
- $M_i(S)$  - miara spójności analizowanego segmentu,
- $N$  - liczba wszystkich pikseli analizowanego segmentu,
- $Im_{in}[x], x \in S$  - kolejne piksele analizowanego segmentu.

Wybrane kryteria łączenia segmentów:

### 1. Różnica średnich:

$$K_1(S_1, S_2) = |M_1(S_1) - M_1(S_2)|, \quad (5.1)$$

$$M_1(S) = \frac{\sum Im_{in}[x]}{N}, \quad x \in S, \quad (5.2)$$

gdzie  $M_1(S)$  - średnia arytmetyczna wszystkich pikseli  $\in S$ .

### 2. Różnica kontrastu:

$$K_2(S_1, S_2) = |M_2(S_1) - M_2(S_2)|, \quad (5.3)$$

$$M_2(S) = MAX(S) - MIN(S), \quad (5.4)$$

gdzie  $M_2(S)$  - kontrast rozumiany jako różnica między największą, a najmniejszą wartością w badanym segmencie.

3. Różnica entropii:

$$K_3(S_1, S_2) = |M_3(S_1) - M_3(S_2)|, \quad (5.5)$$

$$M_3(S) = - \sum P_x * \log_2(P_x), \quad (5.6)$$

gdzie:

$M_3(S)$  - miara entropii pikseli,

$P_x$  - prawdopodobieństwo piksela  $x$  wyliczane ze znormalizowanego histogramu.

4. Różnica wariancji:

$$K_4(S_1, S_2) = |M_4(S_1) - M_4(S_2)|, \quad (5.7)$$

$$M_4(S) = \frac{\sum (Im_{in}[x] - \acute{s}rednia(S))^2}{N}, \quad x \in S, \quad (5.8)$$

gdzie  $M_4(S)$  - wariancja pikseli.

5. Różnica jednolitości:

$$K_5(S_1, S_2) = |M_5(S_1) - M_5(S_2)|, \quad (5.9)$$

$$M_5(S) = 1 - \frac{wariancja(S)}{\acute{s}rednia(S)^2}, \quad (5.10)$$

gdzie  $M_5(S)$  - definicja jednolitości (ang. *uniformity*).

6. Korelacja histogramów:

$$K_6(S_1, S_2) = \frac{H_1 \cdot H_2}{norm(H_1) \cdot norm(H_2)}, \quad (5.11)$$

gdzie:

$H_i$  - histogram segmentu ( $i$ ),

$H_1 \cdot H_2$  - iloczyn skalarny pomiędzy  $H_1$ , a  $H_2$ ,

$norm(H_i)$  - norma euklidesowa histogramu  $H_i$ .

7. Kontrast brzegu:

$$K_7(S_1, S_2) = \frac{1}{M} * \sum \max(C(s, t)), \quad t \in V(s) \wedge t \in S_2 \wedge s \in S_1, \quad (5.12)$$

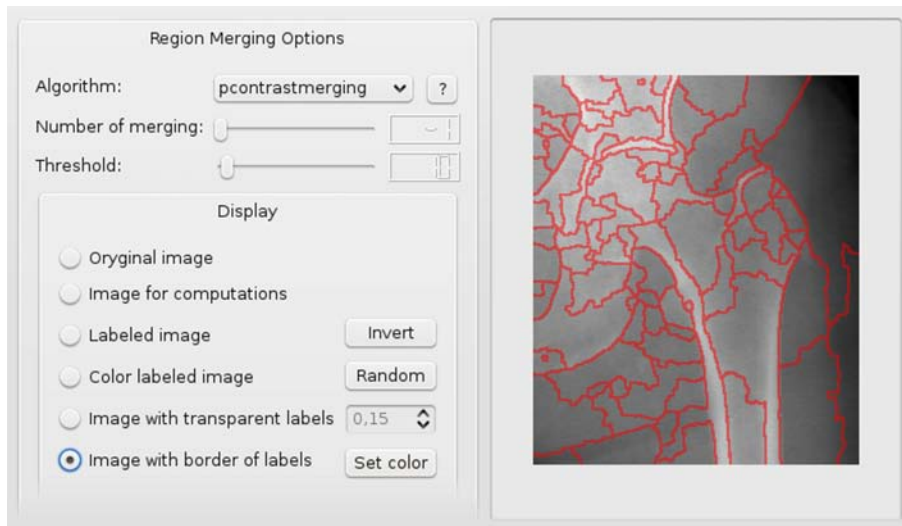
$$C(s, t) = |Im_{in}[s] - Im_{in}[t]| \quad (5.13)$$

gdzie:

$V(s)$  - punkty sąsiednie piksela  $s$ ,

$M$  - liczba pikseli na granicy między segmentem  $S_1$ , a  $S_2$ .

Wszystkie powyższe kryteria łączenia segmentów dostępne są w otwartym analizatorze obrazu *Pandore*<sup>2</sup> [90]. Na potrzeby ich przetestowania przygotowano graficzny interfejs użytkownika<sup>3</sup>, który ułatwił poszukiwanie odpowiedniej wartości granicznej dla poszczególnych kryteriów (Rys. 5.1).



Rysunek 5.1: Testowa aplikacja prezentująca działanie różnych kryteriów łączenia segmentów (*praca własna*).

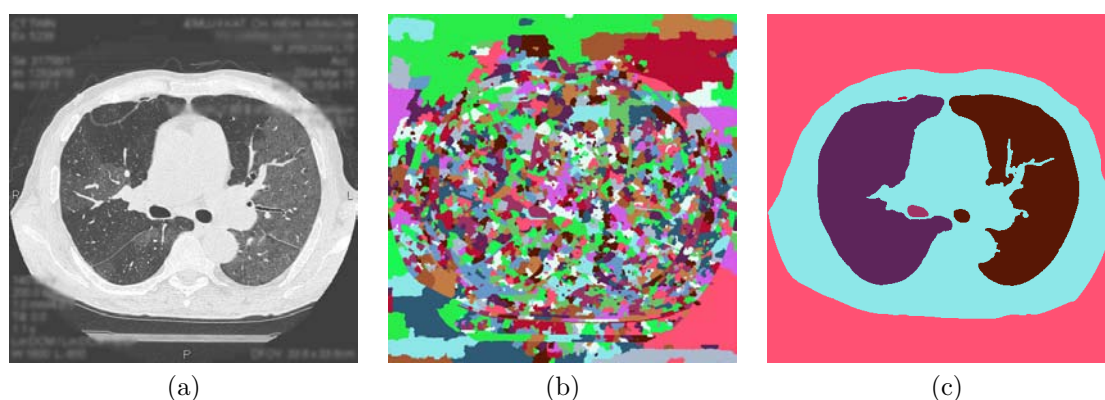
Przetestowane zostały dwa różne typy obrazów: obrazy radiologiczne kości udowej i obrazy tomograficzne klatki piersiowej. Pierwsze z nich charakteryzują się dużą niejednorodnością i problemem z wydzieleniem obszaru kości udowej, natomiast drugie - obrazy klatki piersiowej - nie stwarzają problemów przy segmentacji poszczególnych obiektów. Obrazy te poddano wstępnej segmentacji algorytmem działów wodnych, poprzedzonej operacją rozmycia Gaussa z taką samą wartością parametru  $\sigma = 1$ . Dla obrazów kości udowej wyższa wartość tego parametru prowadziła na niektórych obrazach do błędów w segmentacji obawiającej się m.in. łączeniem fragmentów kości z tłem. Natomiast dla obrazów przedstawiających płuca pomimo tego, że parametr sigma mógłby zostać znacznie zwiększony pozwalając tym samym na uzyskanie dużo większych segmentów, to w celu prezentacji możliwości algorytmu łączenia segmentów został on ustalony na takim samym poziomie.

Na potrzeby omówienia algorytmów łączenia segmentów w kontekście dużych obiektów wybrano po jednym obrazie z każdej serii zdjęć. Dla przykładowego obrazu tomograficznego klatki piersiowej (Rys. 5.2a) wszystkie omówione wyżej kryteria pozwoliły na znaczną redukcję segmentów i doprowadziły do wydzielenia obszaru

<sup>2</sup>Analizator Pandore jest zbiorem programów działających w linii komend, oraz biblioteką C++.

<sup>3</sup>Aplikację wraz z pełnym kodem źródłowym opublikowano na stronie internetowej autora niniejszej pracy pod następującym adresem: <https://sites.google.com/site/rafalpetryniakresearchpl/software/pandore-gui-modules/region-merging>.

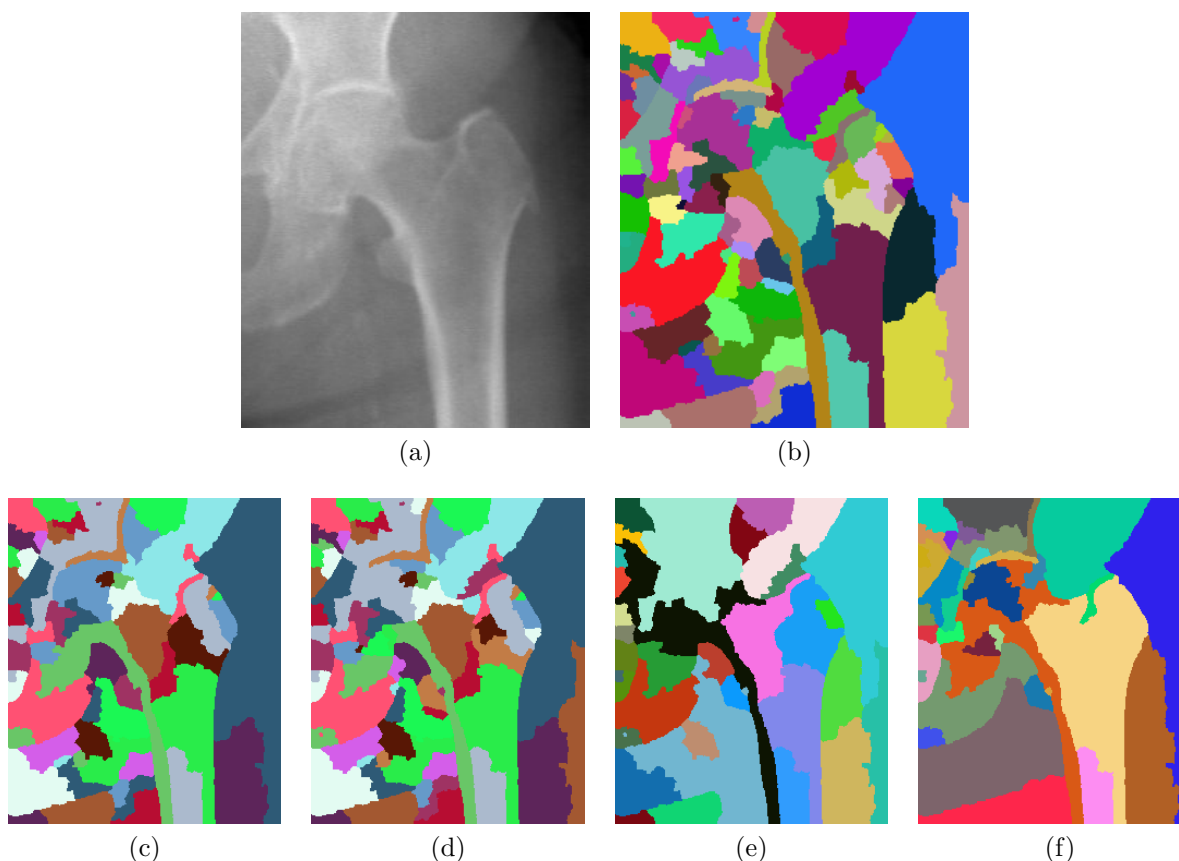
płuc, korpusu ciała i tła (Rys. 5.2c). Wartość graniczna dla każdego kryterium była dobierana indywidualnie poprzez systematyczne jej zwiększanie do momentu uzyskania satysfakcjonujących wyników. Ponieważ wyniki dla każdego kryterium były bardzo zbliżone, dlatego na obrazie 5.2c przedstawiono wyniki tylko dla kryterium bazującego na średniej arytmetycznej (wartość graniczna kryterium=100). Opierając się na zebranych spostrzeżeniach można przypuszczać, że poprawna segmentacja byłaby również możliwa na pozostałych obrazach klatki piersiowej. Jeśli jednak stała wartość graniczna kryterium dla wszystkich obrazów (np. wartość 100 dla średniej arytmetycznej) nie dawałaby dobrych rezultatów, wówczas można rozważyć przygotowanie algorytmu, który proponowałby odpowiednią wartość indywidualnie dla każdego obrazu.



Rysunek 5.2: Wyniki łączenia segmentów dla wybranego obrazu klatki piersiowej: (a) obraz oryginalny, (b) wstępna segmentacja algorytmem działów wodnych, (c) łączenie segmentów z zastosowaniem kryterium bazującego na średniej o wartości granicznej 100 (*opracowanie własne*).

W odróżnieniu od obrazów klatki piersiowej, które w zakresie testowanych kryteriów łączenia segmentów cechowały się dużą jednorodnością, analiza przykładowego zdjęcia kości udowej nie dała oczekiwanych rezultatów. Głównym celem tej analizy była próba wydzielenia kości udowej lub chociażby znaczna redukcja liczby segmentów w jej obszarze. Niestety, szybciej dochodziło do niepoprawnych połączeń segmentów należących do różnych obiektów, aniżeli do scalenia segmentów wewnątrz poszczególnych obiektów. Na rysunku 5.3 przedstawiono wyniki łączenia segmentów dla wybranych kryteriów (dla pozostałych kryteriów uzyskano podobne wyniki) z wartością graniczną, dla której uzyskano pierwsze niepoprawne złączenia. Widać, że już dla niskiej wartości granicznej dla poszczególnych kryteriów szybciej dochodziło do scalenia fragmentów z tłem, niż do oczekiwanych połączeń zwiększających obszary odpowiednich obiektów. Nawet kryterium oparte na kontraście brzegu (Rys. 5.3f), które pozwoliło na uzyskanie stosunkowo dużych segmentów i nie złączyło obszaru kości z tłem, spowodowało błędne scalenie głowy kości

udowej z fragmentem miednicy. Podobne problemy, jak te opisane dla wybranego obrazu, miały również miejsce na pozostałych obrazach przedstawiających kość udową.



Rysunek 5.3: Wyniki łączenia segmentów dla wybranego obrazu kości udowej: (a) obraz oryginalny, (b) wstępna segmentacja algorytmem działów wodnych, łączenie segmentów z zastosowaniem kryterium bazującego na: (c) średniej ( $próg=5$ ), (d) wyrównaniu histogramów ( $próg=0.85$ ), (e) jednolitości ( $próg=0.996$ ), (f) kontraście brzegu ( $próg=2$ ) (*opracowanie własne*).

Podsumowując, stosując popularne kryteria łączenia segmentów trudno wykonać poprawną segmentację dużych obiektów na obrazach niejednorodnych. Obrazy tego typu, tak jak wspomniano już o tym w rozdziale 2, sprawiają trudność w jednoznacznym określeniu cech znajdujących się na nich obiektów. Pomimo istnienia tego typu problemów, w literaturze dotyczącej segmentacji obrazu pojawiło się wiele prac próbujących skutecznie rozwiązać zadanie łączenia regionów (m.in. [4, 35, 48]). Również w niniejszej pracy zostanie zaproponowany nowy algorytm tego typu, dedykowany szczególnie do detekcji lub zmniejszenia liczby segmentów na obrazach przedstawiających duże obiekty. Jego szczegółową analizę zaprezentowano w III części pracy w rozdziale 6.2. [Propozycja algorytmu łączenia regionów z wykorzystaniem informacji o krawędzi.](#)

## Część III

# Algorytmy segmentacji oraz detekcji dużych obiektów



Proces segmentacji obrazu, nawet przy zastosowaniu tak zaawansowanych algorytmów, jak działy wodne lub algorytmy grafowe, może dawać rezultaty inne od oczekiwanych. Z jednej strony może to wynikać z faktu, że analizowany obraz nie jest najlepszej jakości lub obiekty w nim zawarte mają słabo zaznaczone granice. Z drugiej strony, nie wystarczy tylko uruchomić wybrany algorytm segmentacji obrazu i czekać na wyniki, ale należy przekazać do algorytmu informację, które obiekty nas interesują (problem detekcji) i jaki poziom szczegółowości powinien zostać rozpatrzony (problem segmentacji). Sam algorytm nie będzie przecież w stanie domyślić się, czy chcemy uzyskać drobne obiekty na obrazie, czy też skupić się na detekcji dużych obiektów składających się z innych mniejszych obiektów. Problem szczególnie jest widoczny w drugim przypadku, gdzie trzeba określić, które obiekty powinny zostać połączone, by zapobiec nadmiernemu rozdrobnieniu na segmenty. Taką informację można przekazać do algorytmu w postaci modelu reprezentującego wiedzę o poszczególnych obiektach na obrazie. Tego rodzaju wiedza może zawierać np. informację o teksturze, kształcie, granicach i położeniu poszczególnych obiektów. Może być również przedstawiona za pomocą semantycznego opisu obiektów i ich wzajemnych zależności, co może być szczególnie przydatne w automatycznym rozumieniu obrazów (ang. *Automatic Image Understanding*) – szerzej opisane w pracy [70] i [49].

W dalszej części rozprawy zostaną zaproponowane trzy modele, które wg autora mogą mieć szczególne zastosowanie do detekcji i segmentacji dużych obiektów:

- Model bazujący na krawędzi (Rozdz. 6) – zostanie zaproponowany nowy algorytm łączenia regionów z wykorzystaniem wcześniej przygotowanego obrazu krawędziowego, który pozwala na znaczne zredukowanie liczby segmentów.
- Model bazujący na kształcie (Rozdz. 7) – będzie się opierał na technice segmentacji z atlasem (ang. *Atlas-based segmentation*) z zaproponowanym dodatkowo etapem ścisłego dopasowania kształtu do obiektu na obrazie.
- Model bazujący na znacznikach (Rozdz. 8) – zalecany do interaktywnej segmentacji obrazu.

### *Segmentacja obrazu z użyciem modelu bazującego na krawędzi*

Granice zalegania obiektów na obrazach cyfrowych zazwyczaj opisywane są za pomocą binarnego (czarno-białego) obrazu krawędziowego<sup>1</sup>. Taki obraz zawiera bardzo dużo informacji, którą można wykorzystać do jego skutecznej segmentacji. Na obrazach niejednorodnych, dla których nie da się jednoznacznie dobrać progu binaryzacji dla obiektów, wyniki detekcji krawędzi mogą być szczególnie pomocne w procesie rozdzielania obiektów. Jest to oczywiście możliwe w przypadku istnienia wyraźnych różnic koloru lub stopnia szarości pomiędzy sąsiednimi obiektami. W przeciwnym wypadku otrzymamy nieciągłe krawędzie i pojawi się problem z ich poprawnym połączeniem. Zadanie detekcji krawędzi może zostać dodatkowo utrudnione w momencie wystąpienia na obrazie lokalnych zakłóceń czyli szumów. Warto w takiej sytuacji zastosować procedurę wygładzania obrazu. Aby nie stracić przy tym informacji o przebiegu krawędzi zaleca się stosowanie jednego ze znanych filtrów nieliniowych [53, 21] (opisane w 3.3).

## 6.1 Problem nieciągłości krawędzi i sposoby ich łączenia

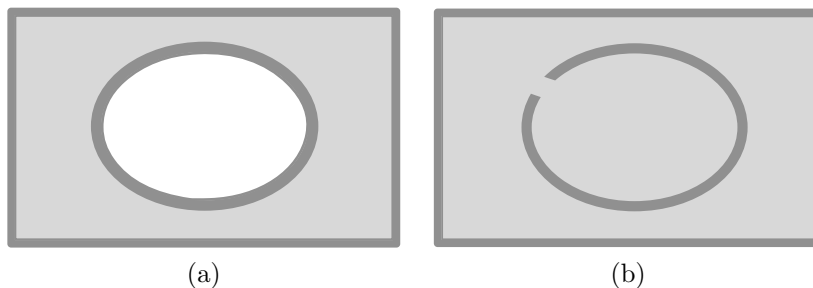
Wykonując segmentację obrazu na podstawie krawędzi bardzo ważne jest, aby były one zamknięte w obrębie poszczególnych obiektów. W przeciwnym wypadku algorytmy indeksujące piksele, które działają na zasadzie łączenia przyległych punktów, połączą również obszary pomiędzy którymi jest niewielka szpara (przykład na rysunku 6.1).

Analizując omówione w dodatku A (strona 115) metody detekcji krawędzi widać, że ten problem występuje dosyć często. Niezbędne w takiej sytuacji może okazać się użycie dodatkowych algorytmów łączących krawędzie. Jedną z najprostszych i skutecznych metod jest zastosowanie progowania z histerezą zamiast użycia jednego, globalnego progu dla wszystkich krawędzi. Podejście to zostało użyte w *algorytmie Canny*<sup>2</sup> i ma duży wpływ na wydłużenie krawędzi obrazu. Niestety, również jego zastosowanie nie daje gwarancji domknięcia wszystkich krawędzi, przez co segmentacja takiego obrazu często kończy się niepowodzeniem (przykład na rysunku 6.2).

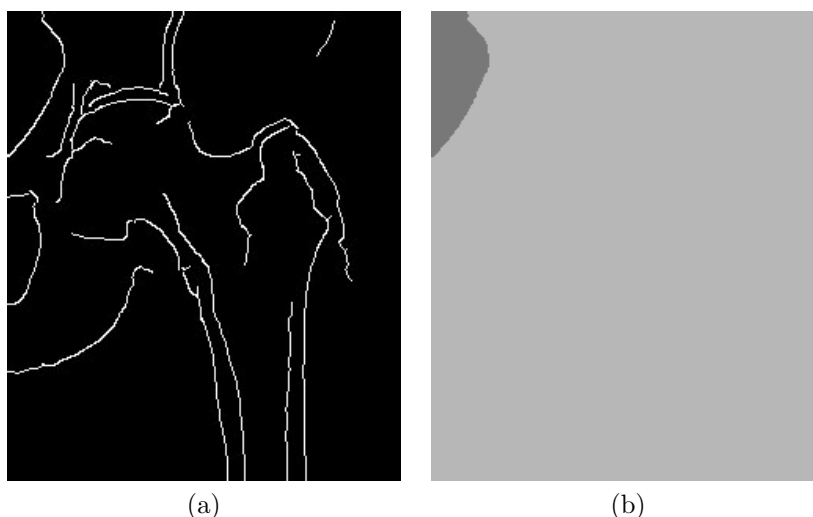
---

<sup>1</sup>Podstawy metod detekcji krawędzi opisano w dodatku A

<sup>2</sup>Podstawy działania *algorytmu Canny* [7] opisano w dodatku A.2.



Rysunek 6.1: Problem segmentacji obrazu z użyciem obrazu krawędziowego. (a) Krawędzie są domknięte i segmentacja przebiega poprawnie. (b) Krawędź elipsy umieszczonej w prostokącie jest otwarta, przez co nie zostaje ona uznana za osobny obiekt (*opracowanie własne*).



Rysunek 6.2: Problem segmentacji obrazu z wykorzystaniem obrazu krawędziowego. (a) Obraz krawędziowy wygenerowany algorytmem Canny [7]. (b) Wynik segmentacji (*opracowanie własne*).

W dalszej części pracy zostaną opisane dwa popularne podejścia, które częściowo rozwiązują problem łączenia krawędzi obrazu i mogą poprawić jakość procesu segmentacji. Dodatkowo na przykładach zostaną pokazane ich słabe strony, które warto mieć na uwadze. Więcej przykładów i technik łączenia krawędzi można znaleźć w książce „*Praktyka analizy obrazu*” [75] w rozdziale 6. „*Analiza struktur komórkowych i ziarnistych*”.

## Śledzenie krawędzi z użyciem obrazu gradientowego

Obraz gradientowy jest wykorzystywany nie tylko do detekcji krawędzi, ale również może być zastosowany do jej uzupełnienia. Przykładowa strategia składa się z następujących kroków:

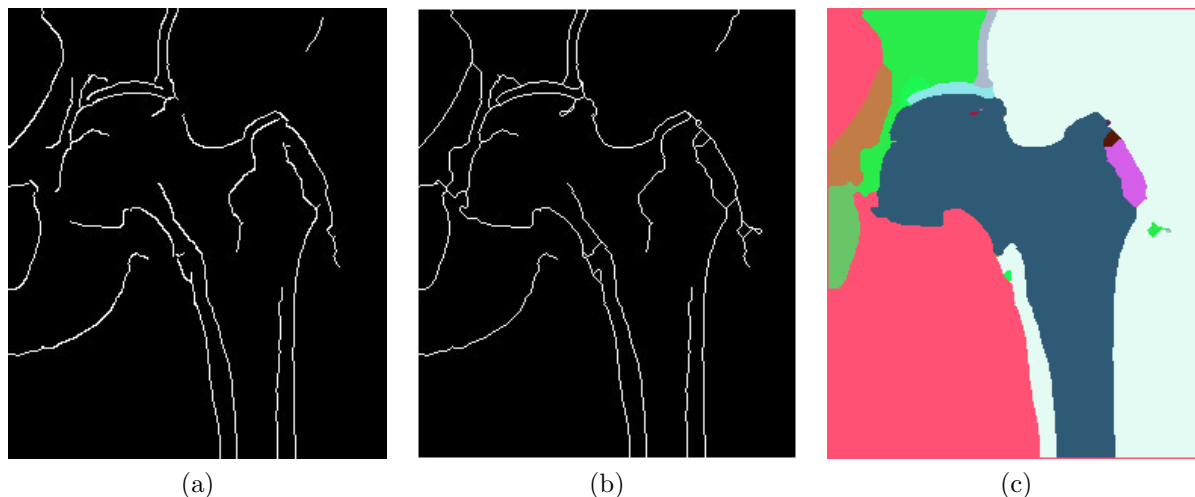
1. Wyznaczenie punktów końcowych dla każdej krawędzi. Punkty te powinny mieć tylko jednego sąsiada, dlatego niezbędne może okazać się zastosowanie szkieletyzacji lub pocieniania (ang. *post-thinning*)<sup>3</sup>.
2. Wybór i dołączenie sąsiada, który ma najwyższą wartość gradientu. Taki punkt staje się następnie punktem końcowym danej krawędzi. Krok ten powtarzamy dopóki:
  - nie trafimy na inną krawędź obrazu, intuicyjnie rozumianą jako przedłużenie obecnej krawędzi (nie mamy jednak pewności czy połączenie będzie poprawne ze względu na rzeczywiste granice obiektów);
  - nie przekroczymy zadanej liczby punktów, która powinna wystarczyć do uzupełnienia krawędzi - zaleca się wprowadzenie tego ograniczenia, ponieważ algorytm w dotychczasowej postaci dokonałby zamknięcia wszystkich linii krawędzi, nawet tam gdzie nie byłoby to wymagane;
  - jeśli okaże się, że wszystkie analizowane punkty sąsiednie mają wartość gradientu równą 0 (brak informacji o krawędzi) anulujemy poprzednio dołączone punkty.

Dodatkowo w tej koncepcji można zawęzić sprawdzanie sąsiadów aktualnego punktu końcowego tylko w wybranych kierunkach, np. w kierunku wskazanym przez poprzedni i bieżący punkt, lub też w kierunkach skierowanych odpowiednio pod kątem  $45^\circ$  i  $-45^\circ$  do tego kierunku.

Dla tak postawionych warunków przykładowe uzupełnienie krawędzi o maksymalnej długości 30 punktów może wyglądać jak na rysunku 6.3. Jak widać, zastosowanie omawianej metody pozwala na znaczne uzupełnienie granic obiektów, szczególnie w miejscach gdzie brakowało kilku punktów na linii krawędzi. Dzięki temu segmentacja obiektów przy użyciu tak przygotowanego obrazu daje znacznie lepsze rezultaty (Rys. 6.3c), aniżeli segmentacja na obrazie krawędziowym bez uzupełnionych krawędzi (Rys. 6.2b). Wadą stosowania tej metody jest brak pewności, że wszystkie krawędzie zostaną uzupełnione. Możliwe jest oczywiście zwiększenie liczby punktów do uzupełnienia na linii krawędzi, ale może to spowodować znaczne zwiększenie liczby podziałów dużych obiektów na mniejsze.

---

<sup>3</sup>Szkieletyzacja powinna zostać przeprowadzona, jeżeli krawędź jest grubości kilku pikseli. Pocienianie natomiast stosujemy w celu upewnienia się, że linia krawędzi będzie miała grubość jednego piksela.



Rysunek 6.3: Uzupełnienie linii krawędzi metodą śledzenia z wykorzystaniem obrazu gradientowego. (a) Obraz krawędziowy. (b) Krawędzie po uzupełnieniu. (c) Segmentacja obrazu na podstawie obrazu (b) (*opracowanie własne*).

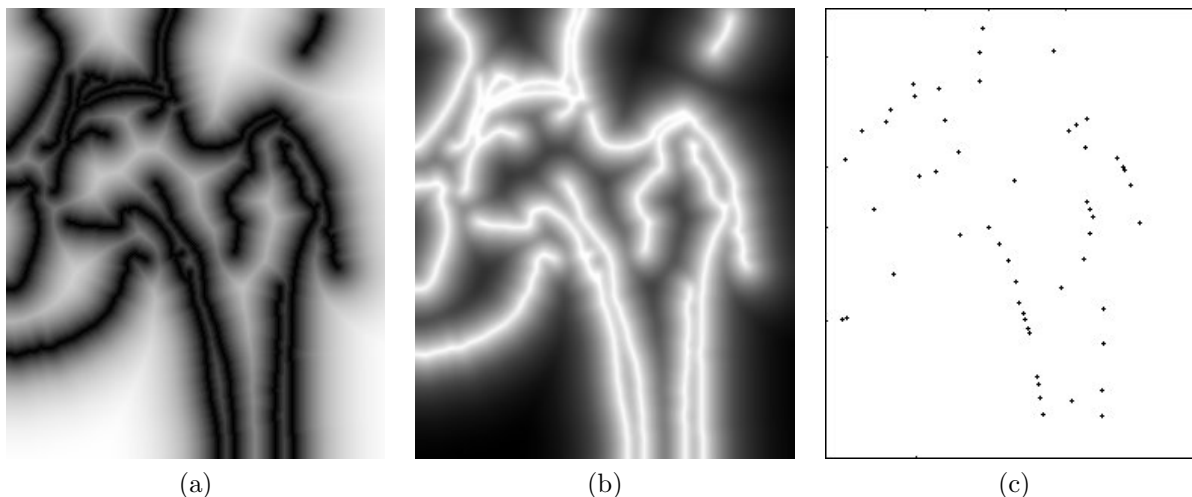
## Zastosowanie algorytmu działów wodnych do łączenia krawędzi obrazu

Innym, często używanym podejściem do łączenia krawędzi na obrazie jest zastosowanie działów wodnych (sposób działania algorytmu opisano w rozdziale 4.1). Jego niewątpliwą zaletą jest fakt, że w wyniku jego działania otrzymamy obraz domkniętych krawędzi. Niestety kosztem tej pewności jest duża liczba drobnych krawędzi skupionych wokół punktów startowych (zazwyczaj lokalnych minimów). Od ich właściwego określenia często zależy dokładność, a co za tym idzie poprawność procesu segmentacji. W celu uruchomienia algorytmu działów wodnych należy przygotować dwa obrazy:

- obraz markerów, który będzie wyznaczał miejsca startowe dla algorytmu i różnicował obszary między sobą (każdy marker po zakończeniu segmentacji będzie reprezentował inny obiekt),
- obraz będący podstawą zalewania poszczególnych obiektów.

Aby zastosować algorytm działów wodnych do łączenia krawędzi, należy pobrać możliwie dużo informacji z samego obrazu krawędziowego. Można przyjąć następujący schemat działania:

1. Przygotowanie mapy odległości, która wskazuje odległość każdego punktu od najbliższej krawędzi. Można ją otrzymać poprzez zsumowanie kolejnych erozji negatywu obrazu krawędziowego (Rys. 6.4a).

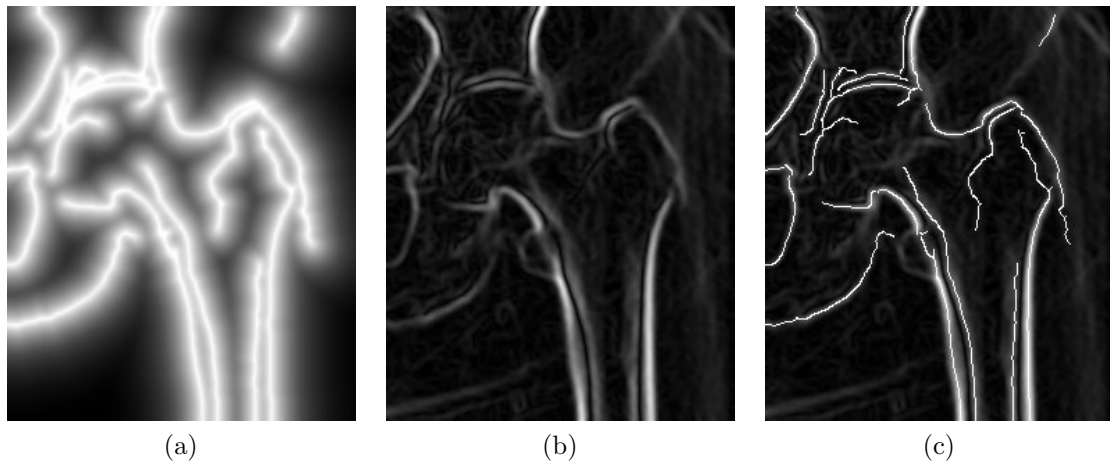


Rysunek 6.4: (a) Mapa odległości, (b) jej negatyw i (c) lokalne minima dla obrazu krawędziowego 6.2a (opracowanie własne).

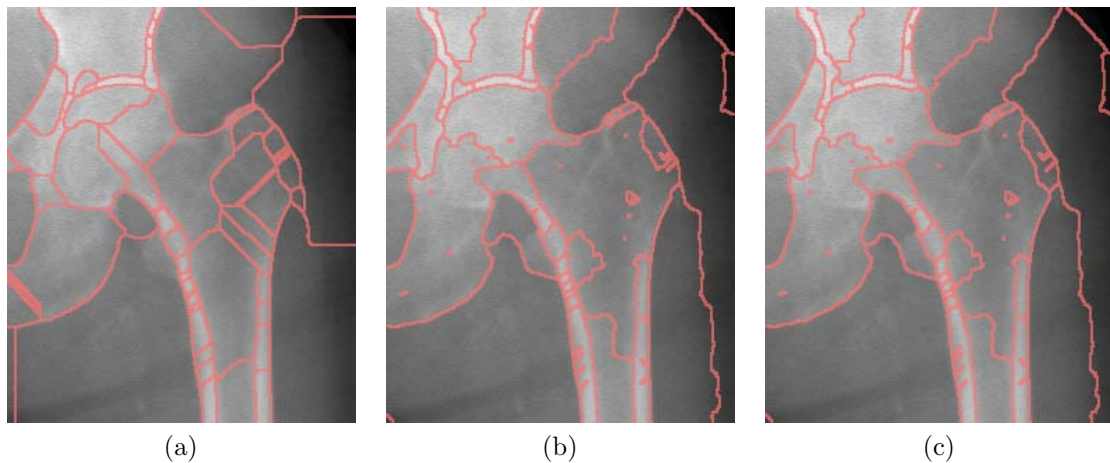
2. Utworzenie negatywu mapy odległości — wymaga tego algorytm działów wodnych, który podąża od punktów ciemniejszych do punktów jaśniejszych (Rys. 6.4b).
3. Wyznaczenie markerów. Mogą to być lokalne minima negatywu mapy odległości (Rys. 6.4c). Jednak aby uniknąć ich nadmiernej liczby, zaleca się zastosowanie wygładzania mapy odległości (np. filtrem Gaussa o wartości  $\sigma = 2$ ).
4. Przygotowanie obrazu służącego do zalewania basenów. Można w tym celu skorzystać z:
  - uprzednio wyznaczonego negatywu mapy odległości (Rys. 6.5a),
  - gradientu oryginalnego obrazu, który rozmyto filtrem Gaussa (Rys. 6.5b),
  - obrazu gradientowego z nałożonymi liniami krawędzi (Rys. 6.5c).

Zastosowanie pierwszego podejścia jest najprostsze i daje dobre rezultaty (Rys. 6.6a), jednak nie gwarantuje zbieżności uzupełnionych krawędzi z rzeczywistymi strukturami na obrazie – dodane linie często mają bardziej regularny charakter, niż byśmy tego oczekiwali. Problem ten może zostać częściowo rozwiązany przy zastosowaniu rozwiązania drugiego (Rys. 6.5b). Uzupełnienie granic z wykorzystaniem gradientu pozwoli na lepsze dopasowanie do struktur obrazu, ale również nie zagwarantuje zachowania wcześniej ustalonych krawędzi (Rys. 6.6b). Pomocne w tym celu może okazać się nałożenie na obraz gradientowy obrazu krawędziowego (Rys. 6.5c), którego zadaniem będzie blokowanie rozlewania się obszarów poza wcześniej wyznaczone krawędzie<sup>4</sup> (Rys. 6.6c).

<sup>4</sup>Wynik działania podejścia trzeciego dla tego obrazu jest niemal identyczny z wynikiem podejścia drugiego. Na innych, analizowanych obrazach kości udowej zdarzało się, że podejście trzecie dawało zauważalnie lepsze rezultaty.



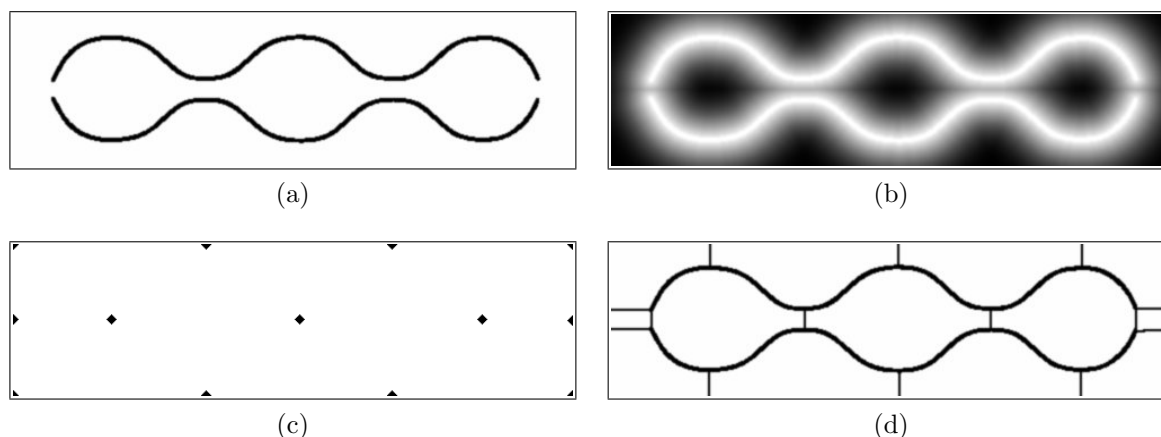
Rysunek 6.5: Różne warianty obrazów będących podstawą zalewania w algorytmie działów wodnych (*opis w tekście, opracowanie własne*).



Rysunek 6.6: Wyniki łączenia krawędzi z wykorzystaniem algorytmu działów wodnych, gdzie obrazem wejściowym był (a) obraz z Rys. 6.5a, (b) obraz z Rys. 6.5b, (c) obraz z Rys. 6.5c (*opracowanie własne*).

Algorytm działów wodnych pomimo tego, że domyka skuteczniej krawędzie niż opisana poprzednio technika śledzenia, nie jest pozbawiony problemów i silnie zależy od wyznaczonych punktów startowych. Wskazana powyżej w punkcie 3 technika wyznaczania markerów na podstawie krawędzi jest bardzo podatna na generowanie nadmiernej liczby punktów startowych, szczególnie w przypadku obiektów o wydłużonych i nieregularnych granicach. Sytuację taką można prześledzić na sztucznie wygenerowanym przykładzie (Rys. 6.7). Przedstawia on wydłużony obiekt z trzema regularnymi wypukłościami, którego linia krawędzi jest przzerwana na jego końcach. Stosując algorytm działów wodnych udaje się te krawędzie połączyć, jednakże dodatkowo zostają wprowadzone nadmiarowe





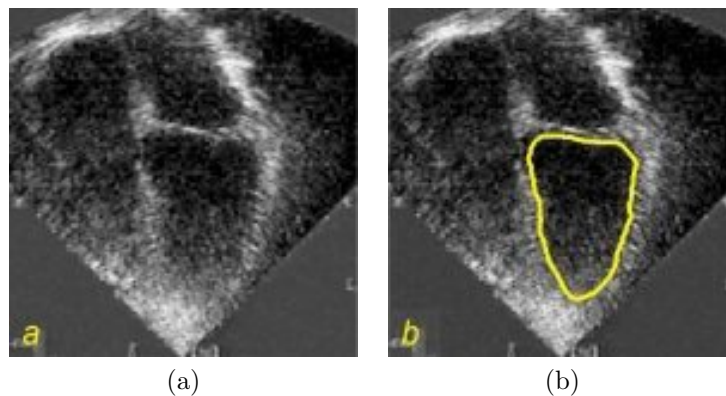
Rysunek 6.7: Prezentacja wykorzystania działów wodnych do zamykania krawędzi dla sztucznego obrazu. (a) Przygotowany obraz wejściowy. (b) Mapa odległości. (c) Lokalne minima. (d) Wynik działania algorytmu działów wodnych (*opracowanie własne*).

podziały zarówno w obszarze obiektu, jak i jego tła (Rys. 6.7d). Wynika to z natury działania operatora generującego mapę odległości (Rys. 6.7b), na podstawie której wyznaczone są punkty startowe do zalewania, będące lokalnymi minimami tej mapy (Rys. 6.7c).

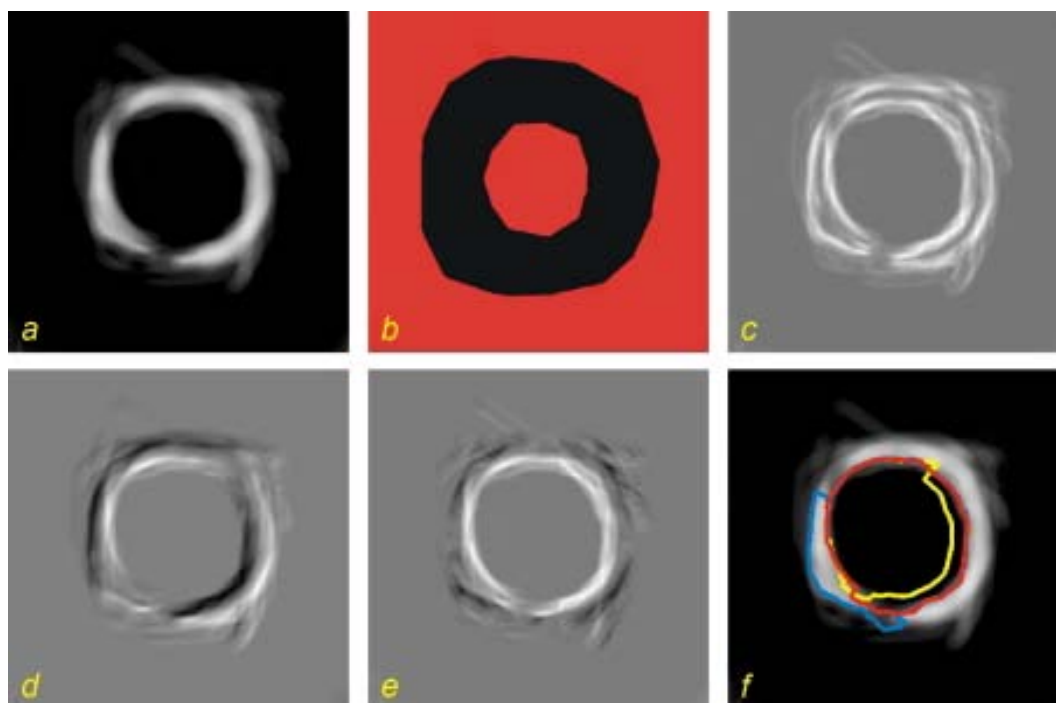
Jak pokazano powyżej, działanie algorytmu działów wodnych jest silnie uzależnione od lokalizacji i liczby punktów startowych. Niestety, nawet wprowadzenie zaawansowanych algorytmów wyznaczających te punkty, lub też zezwolenie użytkownikowi na ich manualne zaznaczenie nie daje nam pewności co do tego, jak zachowa się algorytm. Aby mieć nad nim jeszcze większą kontrolę, a co za tym idzie możliwość uzyskiwania pożądanych efektów, warto również zwrócić uwagę na właściwe przygotowanie obrazu służącego do zalewania. Zazwyczaj w tym celu stosuje się obraz gradientowy uzyskany z obrazu oryginalnego po wcześniejszym wygładzeniu filtrem Gaussa. W przypadku mocnych i ciągłych granic podejście to daje dobre rezultaty. Jednakże w miejscach gdzie krawędzie zanikają, tradycyjny gradient będący złożeniem różnicowania punktów w kierunku pionowym i poziomym może nie być optymalnym rozwiązaniem.

Jak pokazano w pracy doktorskiej Zbigniewa Latały [34] odpowiednie przygotowanie obrazu do zalewania może być kluczowe w uzyskaniu dobrego wyniku. Autor pracy, o której mowa, analizował problem detekcji lewej komory serca na obrazach echokardiograficznych. Obrazy te zazwyczaj są bardzo słabej jakości (przykład na rysunku 6.8a) i stwarzają istotne problemy w trakcie ich komputerowej analizy. Problem detekcji udało się autorowi rozwiązać poprzez zastosowanie sekwencyjnego generowania znacznika ograniczającego obszar lewej komory serca, jak również dzięki wprowadzeniu *gradientu promieniowego*. Gradient ten w odróżnieniu od tradycyjnego gradientu





Rysunek 6.8: (a) Przykładowy obraz echokardiograficzny serca oraz (b) wyniki detekcji lewej komory serca z wykorzystaniem gradientu promieniowego (na podstawie [34]).



Rysunek 6.9: Ilustracja wpływu obrazu gradientowego na wynik detekcji działów wodnych: (a) obraz wejściowy, (b) znacznik do detekcji działów wodnych, (c) gradient morfologiczny, (d) gradient definiowany przez użytkownika, (e) gradient promieniowy, (f) wyniki detekcji z użyciem poszczególnych obrazów gradientowych; niebieski, żółty i czerwony odpowiednio dla obrazów gradientowych c, d oraz e (na podstawie [34]).

może być wyznaczany w różnych kierunkach i istnieje możliwość określania punktu obrazu względem, którego będzie on wyliczany. Porównanie działania algorytmu działów wodnych z wykorzystaniem gradientu promieniowego i innych popularnych form tego operatora zilustrowano w omawianej pracy za pomocą rysunku 6.9. Dobre rezultaty, które zaznaczono kolorem czerwonym na rysunku 6.9f zostały również potwierdzone na kilku testowych seriach obrazów echokardiograficznych serca (przykład na rysunku 6.8b).

## 6.2 Propozycja algorytmu łączenia regionów z wykorzystaniem informacji o krawędzi

Aby obraz krawędziowy mógł służyć jako podstawa procesu segmentacji obrazu, niezwykle istotne jest właściwe domknięcie przerwanych krawędzi. Nie zawsze jednak jest to możliwe w sposób automatyczny i może się zdarzyć, że w wyniku stosowania technik domykania krawędzi otrzymamy nie tylko nadmiarowe, ale nawet fałszywe krawędzie poprowadzone przez algorytm tam, gdzie tego byśmy się nie spodziewali. Dla przykładu, taka sytuacja może mieć miejsce w przypadku stosowania algorytmów grafowych i podania zbyt dużej wartości minimalnej dla poszukiwanej ścieżki.

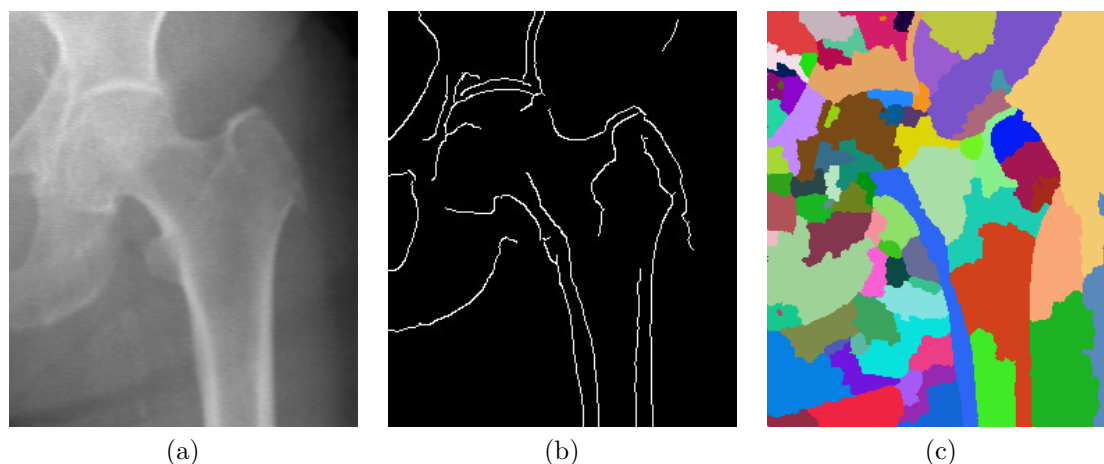
W tym rozdziale zostanie zaprezentowany nowy algorytm łączenia regionów, który będzie opierał się na wcześniej przygotowanym obrazie krawędziowym z wykorzystaniem *algorytmu Canny* oraz obrazie po wstępnej segmentacji, gdzie zalecanym algorytmem tworzącym segmenty jest *algorytm działów wodnych*. W literaturze dotyczącej analizy obrazu znane są już koncepcje łączenia regionów bazujące na analizie granic między segmentami: [6, 26]. W najprostszym przypadku, ich użycie sprowadza się do przygotowania obrazu gradientowego i wyliczenia średniej wartości punktów dla poszczególnych granic segmentów. Tam gdzie ta wartość jest niższa od zadanej wartości progowej – występuje słaba linia krawędzi lub jej brak – segmenty są łączone w jeden obiekt. Podejście to daje dobre rezultaty w przypadku istnienia wyraźnych granic między obiektami, natomiast tam gdzie występują długie i nieregularne krawędzie, może się okazać, że segmenty zostaną błędnie sklasyfikowane jako jeden obiekt.

Celem proponowanego algorytmu będzie znaczna redukcja liczby segmentów składowych dużych obiektów na obrazach niejednorodnych, a warunkiem działania możliwość wyznaczenia przeważającej liczby krawędzi. Ogólna koncepcja algorytmu wraz z opisem przykładowej realizacji zostaną omówione w kolejnych podrozdziałach.

### Koncepcja algorytmu

Zastosowanie opisanych w rozdziale 4 technik segmentacji obrazu często powoduje znaczne rozdrobnienie dużych obiektów na wiele małych segmentów. Techniki te oczywiście mają możliwość ustawienia czułości detekcji, ale bywa to na tyle kłopotliwe, że w praktyce często korzysta się z algorytmów łączenia regionów z odpowiednio dobranym kryterium jednorodności np. wartość średnia, mediana, wariancja, korelacja histogramu i inne (zostało to szerzej opisane w rozdziale 5). Jednak w przypadku kiedy analizowany obiekt nie jest jednorodny i składa się z kilku różnych obiektów pojawia się problem, jak przekazać niezbędną wiedzę do algorytmu, które segmenty powinny zostać ze sobą połączone. W takiej sytuacji szansą na poprawną segmentację może okazać się

informacja o krawędziach tych obiektów. Stosując zaawansowane algorytmy, takie jak opisany w dodatku A.2 *algorytm Canny*, można otrzymać wyraźny zarys głównych obiektów obrazu i złagodzić wpływ szumu na wynik detekcji. Nawet jeśli krawędzie, które otrzymamy nie będą ciągłe, to warto rozważyć ich użycie do rekonstrukcji dużych obiektów z wielu mniejszych segmentów, lub przynajmniej do ograniczenia liczby podziałów tych obiektów.

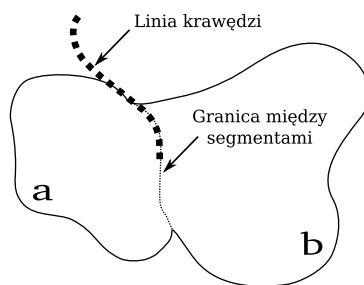


Rysunek 6.10: Obraz rentgenowski kości udowej: obraz oryginalny (a) i obrazy wejściowe do algorytmu łączenia segmentów: (b) Krawędzie *Canny* – zastosowano rozmycie Gaussa o wartości  $\sigma = 2$ , (c) Działy wodne – zastosowano rozmycie Gaussa o wartości  $\sigma = 1$ , (opracowanie własne).

Proponowany algorytm ma za zadanie łączenie tych regionów, które znajdują się wewnątrz obszarów zakreślonych przez linie krawędzi. Aby to osiągnąć, należy łączyć ze sobą poszczególne segmenty według określonej kolejności, dopóki nie zostanie przekroczona zadana przez użytkownika wartość graniczna ( $K_g$ ). Przygotowane w tym celu **kryterium łączenia** ( $K$ ) szereguje pary segmentów, począwszy od tych, które mają najdłuższą wspólną granicę i są w możliwie małym stopniu (najlepiej wcale) rozdzielone linią krawędzi. Jego szczegółowe wyznaczenie polega na zliczeniu punktów na granicy sąsiadujących ze sobą segmentów (Rys. 6.11), odjęciu od tej wartości liczby punktów należących do krawędzi i podzieleniu przez obwód każdego segmentu z osobna (wzory 6.1 i 6.2).

$$K_{ab} = \frac{dg_{ab} - dk_{ab}}{Op_a} \quad (6.1)$$

$$K_{ba} = \frac{dg_{ba} - dk_{ba}}{Op_b} \quad (6.2)$$



Rysunek 6.11: Pojęcie granicy oraz krawędzi między dwoma segmentami (*opracowanie własne*).

gdzie:

$K_{ab}, K_{ba}$  - kryterium łączenia segmentu  $a$  z segmentem  $b$  i na odwrót

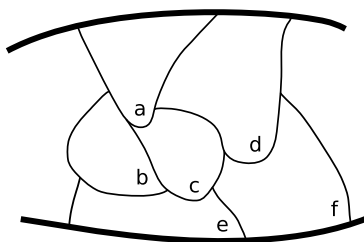
$dg_{ab} = dg_{ba}$  - długość granicy pomiędzy segmentami  $a$  i  $b$

$dk_{ab} = dk_{ba}$  - długość linii krawędzi na granicy segmentu  $a$  i  $b$

$Op_a, Op_b$  - obwód segmentu  $a$  i obwód segmentu  $b$

Obliczone w ten sposób wartości klasyfikowane są dodatkowo w zależności od występowania krawędzi na obwodzie łączonych segmentów<sup>5</sup> (Rys. 6.12):

- grupa 3 - żaden z segmentów nie styka się z linią krawędzi (segment  $b$  i  $c$ )
- grupa 2 - jeden z segmentów dotyka krawędzi (segment  $a$  i  $b$ )
- grupa 1 - oba segmenty stykają się z linią krawędzi (segment  $d$  i  $f$ ).



Rysunek 6.12: Klasyfikacja granic segmentów ze względu na sąsiedztwo krawędzi (*opis w tekście, opracowanie własne*).

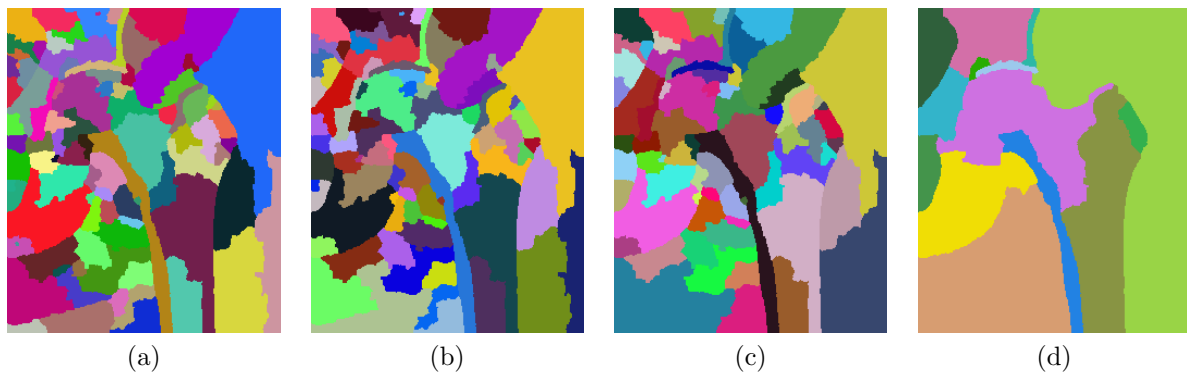
Następnie wszystkie elementy o wartości wyższej od wartości progowej<sup>6</sup> zadanej przez użytkownika  $K_g$  umieszczane są w kolejce priorytetowej. Ostatecznie uruchamiany jest algorytm łączenia regionów, który rozpoczyna swoje działanie począwszy od elementów

<sup>5</sup>Grupy numerowane są w kolejności malejącej ze względu na kolejkę priorytetową, do której będą dodawane wartości kryterium, a która jest sortowana w kolejności malejącej.

<sup>6</sup>Wartość progowa kryterium łączenia segmentów w proponowanym algorytmie wpływa na poziom szczegółowości procesu segmentacji. Im ta wartość jest wyższa tym mniej segmentów zostanie połączone. Dla uproszczenia przyjęto podawanie tej wartości w procentach.

o najwyższej wartości z grupy trzeciej, a skończywszy na elementach najmniejszych z grupy pierwszej. Po każdym kroku algorytmu obliczane są ponownie wartości kryteriów łączenia dla nowo powstałego segmentu i wszystkich jego sąsiadów.

Wynik działania algorytmu można prześledzić na rysunku 6.13. Zostały na nim również przedstawione wyniki pośrednie po zakończeniu analizy poszczególnych grup segmentów.



Rysunek 6.13: Wynik końcowy (d) i wyniki pośrednie proponowanego algorytmu łączenia segmentów. (a) Wejściowy obraz segmentów. Wyniki po połączeniu granic należących do grupy 1 (Rys. b), 2 (Rys. c) i 3 (Rys. d) (*opracowanie własne*).

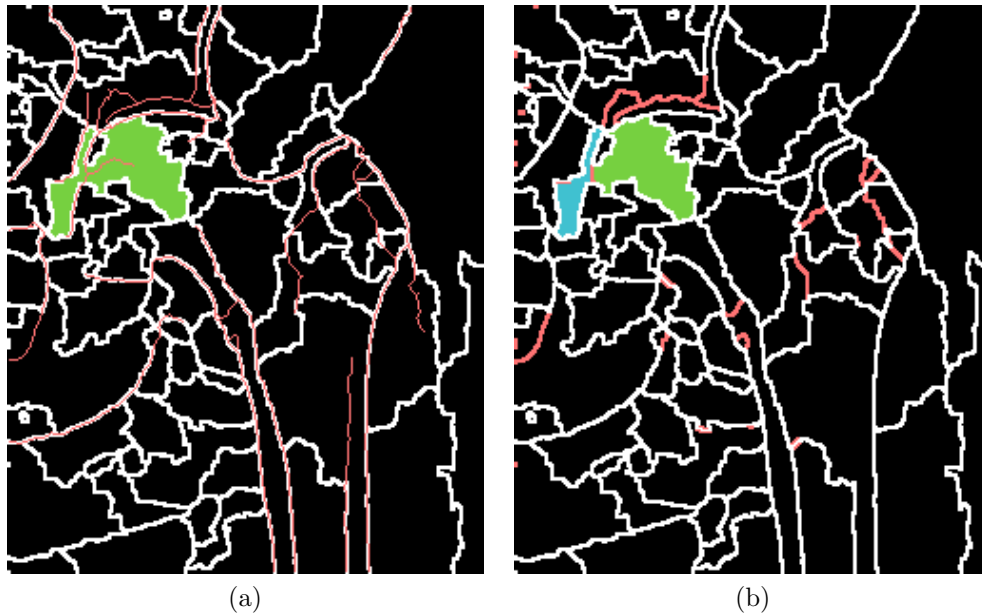
## Realizacja algorytmu

W celu dokładnego przeanalizowania proponowanego algorytmu, a co za tym idzie skutecznego wykorzystania go w praktyce, szczegółowo zostaną opisane etapy jego realizacji. Przedstawiono również dodatkowe zalecenia mogące ułatwić implementację algorytmu oraz odpowiednie przygotowanie danych do analizy.

### Przygotowanie obrazów do analizy

W zdecydowanej większości przypadków linie detekcji krawędzi *Canny* są zgodne z liniami działów wodnych. Wynika to z faktu, iż podstawą działania obu algorytmów jest poszukiwanie krawędzi znajdujących się możliwie blisko grzbietu gradientu. Oczywiście algorytm działów wodnych tworzy zamknięte obszary – zatem linii krawędzi/podziałów może być znacznie więcej niż tych wygenerowanych przez *algorytm Canny*. Może się jednak zdarzyć tak, że linie krawędzi *Canny* będą wchodzić, a nawet przecinać segmenty działów wodnych. Aby zachować spójność informacji przekazywanej przez te dwa algorytmy zaleca się, aby **segmenty, przez które w całości przechodzi linia krawędzi zostały podzielone na części**. Problem tego typu jest widoczny na rysunku 6.14a. Segment wyróżniony kolorem zielonym ma długą i nieregularną powierzchnię, przez którą

przechodzi linia krawędzi. Zachowanie tego segmentu w oryginalnej postaci mogłoby znacznie zakłócić algorytm łączenia, powodując scalenie segmentów leżących na jego krańcach. Obraz po podziale jest widoczny na rysunku 6.14b.



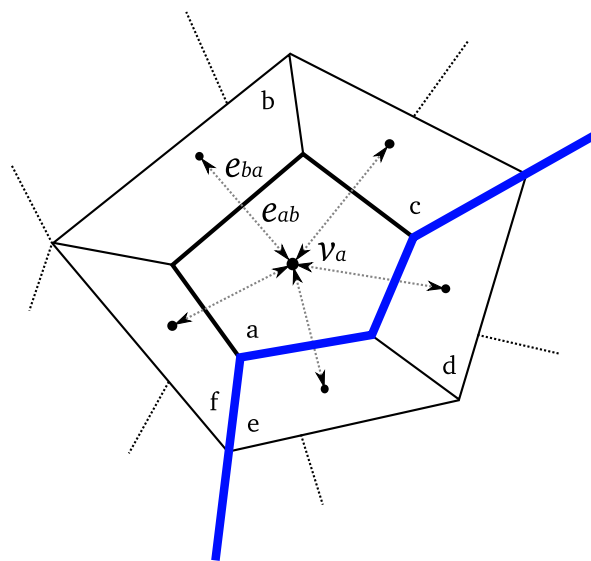
Rysunek 6.14: Przykład niespójności wyników algorytmu działów wodnych i detekcji krawędzi algorytmem *Canny*. (a) Czerwoną linią zaznaczono linie krawędzi *Canny*, białą natomiast działły wodne. (b) Segment zielony z rysunku (a) zostaje podzielony na dwie części (*opis w tekście*). Linia czerwona wskazuje nowe linie krawędzi dodane w miejscach, gdzie linia *Canny* przecina w całości dany segment (*opracowanie własne*).

Opisywany proces podziału może zostać zrealizowany za pomocą następującej sekwencji operacji:

1. Nieznaczne domknięcie krawędzi na obrazie krawędziowym *Canny* umożliwi ewentualne przecięcie tych segmentów, przez które przechodzi linia krawędzi w znacznym stopniu - zaleca się zastosowanie algorytmu śledzenia krawędzi z użyciem obrazu gradientowego (opisany w rozdziale 6.1) o długości do 10 pikseli (wynik tej operacji jest widoczny na rysunku 6.14a jako czerwona linia),
2. Dodanie obrazu działów wodnych i obrazu krawędziowego - powstanie wspólny obraz krawędzi z obu algorytmów,
3. Użycie algorytmu usuwania gałęzi (ang. *barb removal*) do odcięcia linii krawędzi, które częściowo wchodzą w obszary segmentów.

### Algorytm łączenia segmentów

Opisana wyżej koncepcja algorytmu (Rozdz. 6.2) zostanie teraz poddana bardziej szczegółowej analizie. Pomocny w tym celu będzie uproszczony schemat<sup>7</sup> podziału obszaru na mniejsze segmenty widoczny na rysunku 6.15. Obraz ten przedstawia sześć segmentów ( $a, b, c, d, e, f$ ) rozdzielonych linią krawędzi *Canny* (pogrubiona, niebieska linia), co może sugerować, że mamy do czynienia z dwoma obiektami - po lewej i po prawej stronie linii krawędzi. Kluczowe z punktu widzenia kryterium łączenia segmentów poszukiwanie najdłuższych granic, przez które nie przechodzi linia krawędzi, pozwala nam łatwo określić, że w tym przykładzie będzie to granica pomiędzy segmentem  $a$  i segmentem  $b$  i to właśnie ona będzie podstawą dalszej analizy<sup>8</sup>.



Rysunek 6.15: Przykładowy schemat podziału segmentów wyrażony w postaci grafu – stan pierwotny (*opracowanie własne*).

#### Przygotowanie grafu opisującego segmenty i granice między nimi

Na potrzeby algorytmu łączenia segmentów obraz po segmentacji jest reprezentowany w postaci grafu symetrycznego<sup>9</sup> zapisanego za pomocą listy sąsiedztwa  $N$ . Dzięki temu każdy z segmentów będzie miał wiedzę o swoich sąsiadach, a krawędź grafu  $e$  łącząca poszczególne wierzchołki  $v$  będzie przechowywać informację o liczbie punktów na granicy między nimi.

<sup>7</sup>Analiza rzeczywistego obrazu z powodu dużej liczby segmentów mogłaby znacznie utrudnić prezentację poszczególnych etapów algorytmu.

<sup>8</sup>Schemat postępowania zostanie przedstawiony dla jednej granicy. Dla pozostałych granic schemat postępowania jest ten sam.

<sup>9</sup>Graf, w którym jeśli istnieje krawędź  $(v_a, v_b)$ , to również istnieje krawędź  $(v_b, v_a)$ .



Z punktu widzenia segmentów  $a$  i  $b$  zostaną zgromadzone następujące informacje:

- listy sąsiedztwa:

$$N_a = \{v_b, v_c, v_d, v_e, v_f\}$$

$$N_b = \{v_a, v_c, v_f\}$$

- krawędzie grafu zawierające informacje o liczbie punktów na granicy segmentów ( $gd$ ) oraz liczbę punktów tej granicy leżących na linii *Canny* ( $dk$ )<sup>10</sup>:

$$e_{ab} = [gd_{ab}, dk_{ab}]$$

$$e_{ba} = [gd_{ba}, dk_{ba}]$$

- węzły grafu przechowujące liczbę punktów na obwodzie segmentów ( $Op$ ) oraz liczbę punktów obwodu z pominięciem punktów, które łączą się z linią krawędzi *Canny* ( $Obk$ ):

$$v_a = [Op_a, Obk_a]$$

$$v_b = [Op_b, Obk_b]$$

### Wyliczenie kryterium łączenia dla wszystkich krawędzi, klasyfikacja i umieszczenie w kolejce priorytetowej

Kiedy mamy już przygotowany zbiór wszystkich krawędzi grafu  $E$ , dla każdej z nich liczymy kryterium łączenia  $K$  korzystając ze wzorów 6.1 i 6.2. W kolejnym etapie każda z krawędzi  $e$  zostanie przydzielona do jednej z grup, zgodnie z zaleceniami podanymi w koncepcji algorytmu (Rozdz. 6.2). Bardziej szczegółowe warunki przydziału do grupy przedstawiono poniżej na przykładzie segmentów  $a$  i  $b$ :

$$g_{ab} = \begin{cases} 3 & \text{dla } \frac{Obk_a}{Op_a} = 1 \text{ i } \frac{Obk_b}{Op_b} = 1 \\ 2 & \text{dla } \frac{Obk_a}{Op_a} = 1 \text{ i } \frac{Obk_b}{Op_b} \neq 1 \\ 1 & \text{dla } \frac{Obk_a}{Op_a} \neq 1 \text{ i } \frac{Obk_b}{Op_b} \neq 1 \end{cases}, \quad g_{ba} = \begin{cases} 3 & \text{dla } \frac{Obk_b}{Op_b} = 1 \text{ i } \frac{Obk_a}{Op_a} = 1 \\ 2 & \text{dla } \frac{Obk_b}{Op_b} = 1 \text{ i } \frac{Obk_a}{Op_a} \neq 1 \\ 1 & \text{dla } \frac{Obk_b}{Op_b} \neq 1 \text{ i } \frac{Obk_a}{Op_a} \neq 1 \end{cases}$$

Kiedy mamy już wyznaczone kryterium łączenia segmentów oraz sklasyfikowaną granicę ze względu na sąsiedztwo krawędzi *Canny*, dodajemy te wartości do kolejki priorytetowej, która jest sortowana malejąco. Dodatkowo w kolejce zapisujemy numery segmentów, których dotyczy wyliczone kryterium:

$$\text{kolejka} [grupa, kryterium, segment_1, segment_2] \leftarrow [gd_{ab}, K_{ab}, a, b] \leftarrow [gd_{ba}, K_{ba}, b, a]$$

<sup>10</sup>Informacja zawarta w  $e_{ab}$  i  $e_{ba}$  jest taka sama, ponieważ wartości  $dg_{ab}$  i  $dg_{ba}$  oraz  $dk_{ab}$  i  $dk_{ba}$  są równe. Przechowywanie jej w powielonej postaci ułatwia jednak implementację algorytmu, ponieważ dla poszczególnych krawędzi grafu  $e$  są liczone wartości kryterium łączenia  $K$ , np. dla krawędzi  $e_{ab}$  jest wyznaczane kryterium  $K_{ab}$ , które różni się od kryterium  $K_{ba}$ .



### Łączenie segmentów

Po uzupełnieniu kolejki informacjami o wszystkich granicach segmentów i jej posortowaniu, zostaje uruchomiona iteracyjna procedura łącząca segmenty, działająca dopóki kolejka nie będzie pusta. Wykonywane w niej są następujące operacje:

1. Pobranie pierwszego elementu z kolejki (element z najwyższej grupy o najwyższej wartości kryterium łączenia  $K$ ). Odczytanie numerów segmentów. Dla omawianego przykładu w pierwszej iteracji będą to następujące wartości:

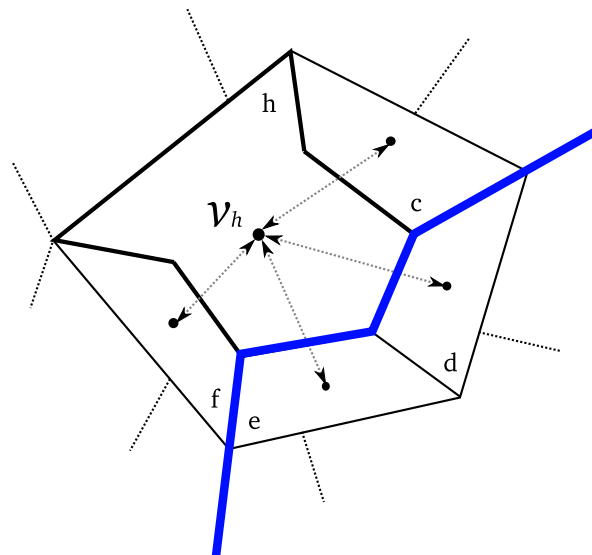
$$segment_1 \leftarrow a$$

$$segment_2 \leftarrow b$$

2. Połączenie segmentów  $segment_1$  i  $segment_2$  w jeden segment i przydzielenie dla niego kolejnego identyfikatora (po połączeniu segmentów  $a$  i  $b$  powstanie segment  $h$ , co zilustrowano na rysunku 6.16).
3. Wyznaczenie liczby punktów na obwodzie nowego segmentu.
4. Przekazanie informacji o połączeniu segmentów  $segment_1$  i  $segment_2$  do ich sąsiadów w celu aktualizacji list sąsiedztwa.
5. Ponowne wyznaczenie kryterium łączenia dla nowego segmentu na wszystkich granicach (zmianie uległ obwód).
6. Wyznaczenie kryterium łączenia  $K$  dla sąsiednich segmentów na granicy z nowym segmentem (obwód sąsiadów nie uległ zmianie, ale zmieniła się długość granicy łączącej te segmenty z nowym segmentem<sup>11</sup>).
7. Dla wszystkich nowo policzonych wartości kryteriów należy wyznaczyć przydział do grupy ze względu na sąsiedztwo krawędzi *Canny*.
8. Dodanie do kolejki wartości kryteriów, które są wyższe od zadanej wartości progowej  $K_g$ . Dołączenie informacji o grupie i numerach segmentów.
9. Zapamiętanie numerów segmentów  $segment_1$  i  $segment_2$  na liście elementów nieaktualnych. Przy ponownym pobraniu elementu z kolejki będą pomijane te z nich, których wartość zmiennej  $segment_1$  lub  $segment_2$  znajduje się na tej liście. Dla omawianego przykładu granice  $ac, ca, af, fa, bc, cb, bf, fb$  nie będą już analizowane.

---

<sup>11</sup>Dotyczy to segmentów, które sąsiadowały z obydwoma połączonymi segmentami. Na omawianym przykładzie 6.16 sytuacja ta odnosi się do segmentu  $c$  i  $f$ .



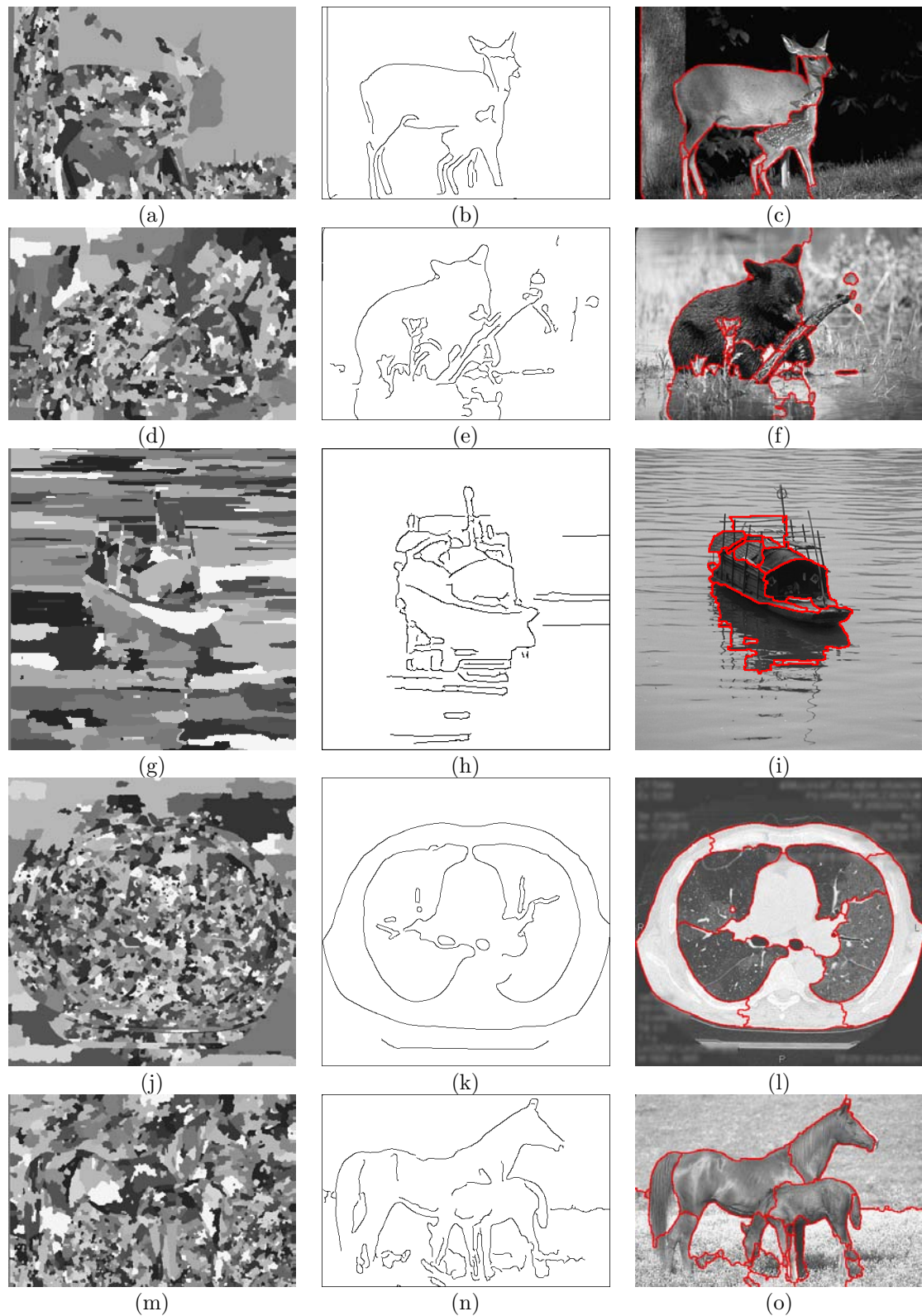
Rysunek 6.16: Przykładowy schemat podziału segmentów wyrażony w postaci grafu – stan po złączeniu segmentu *a* oraz segmentu *b* w segment *h* (opracowanie własne).

### Analiza rezultatów i wnioski

Dla obrazu radiologicznego kości udowej (Rys. 6.10a) głównym celem było zmniejszenie liczby segmentów w obrębie samej kości udowej. Wartość progowa kryterium  $K_g$  ustalona na poziomie 15% pozwoliła na poprawne złączenie segmentów w obszarze tła oraz na redukcję segmentów w obrębie kości (Rys. 6.13d). Próby zmniejszenia tej wartości w celu zmniejszenia liczby segmentów prowadziły do błędnych połączeń na niektórych zdjęciach kości udowej, na których testowano algorytm<sup>12</sup>. Wyniki działania proponowanego algorytmu dla innych przykładów można zobaczyć na rysunku 6.17. Aby skupić uwagę na większych szczegółach (dużych obiektach), wartość parametru *sigma* algorytmu detekcji krawędzi *Canny* ustawiono na 3. Pozwoliło to na usunięcie drobnych krawędzi obrazu i ostatecznie miało istotny wpływ na dokładność wyników algorytmu łączenia segmentów. Analizując przykłady z rysunku 6.17 warto zwrócić szczególną uwagę na następujące wyniki działania algorytmu:

- obraz „sarny” (Rys. 6.17a, 6.17b i 6.17c) - pomimo tego, że na obrazie krawędziowym zwierzęta są złączone ze sobą, to ostatecznie zostały one poprawnie rozdzielone na dwa duże obiekty,
- obraz „niedźwiadek” (Rys. 6.17d, 6.17e i 6.17f) - niedźwiadek i jego odbicie w wodzie zostały poprawnie zaznaczone jako różne obiekty,

<sup>12</sup>Prezentowany w tym rozdziale algorytm łączenia segmentów wykorzystano na potrzeby optymalizacji omawianego w kolejnym rozdziale algorytmu detekcji obiektów. Na rysunku 7.11 (str. 83) można zobaczyć jego działanie dla kilku innych obrazów kości udowej i innych obrazów radiologicznych.



Rysunek 6.17: Przykłady działania algorytmu łączenia segmentów na podstawie krawędzi dla innych zdjęć. Lewa kolumna: segmenty działów wodnych (rozmycie Gaussa,  $\sigma = 1$ ). Środkowa kolumna: Krawędzie *Canny* (rozmycie Gaussa,  $\sigma = 3$ ). Prawa kolumna: kontur wynikowych segmentów naniesiony na obraz oryginalny (*opracowanie własne*).

- obraz „statek” (Rys. 6.17g, 6.17h i 6.17i) - drobne elementy widoczne na obrazie krawędziowym zostały pominięte,
- obraz „płuca” (Rys. 6.17j, 6.17k i 6.17l) - pomimo tego, że krawędzie poszczególnych obiektów są w znacznej mierze ciągłe, to ze względu na ich wydłużony kształt (szczególnie korpus ciała) zostały one podzielone na mniejsze części,
- obraz „konie” (Rys. 6.17m, 6.17n i 6.17o) - liczba segmentów w obrębie zwierząt została znacznie zredukowana, jednak zarówno tylne nogi większego zwierzęcia, jak i część tułowia mniejszego zostały błędnie oznaczone.

Wynik działania proponowanego algorytmu silnie zależy od właściwego wyznaczenia krawędzi obrazu. Dlatego na obrazach, gdzie granica między obiektami jest słabo widoczna, może on nie działać poprawnie. W takich sytuacjach można dodatkowo uwzględnić omawiane w rozdziale 5.2 inne znane kryteria łączenia regionów, biorące pod uwagę również charakterystykę wartości pikseli wewnątrz segmentów. Umożliwiłoby to uniknięcie sytuacji łączenia dwóch różnych obiektów, pomiędzy którymi była zbyt duża przerwa w krawędzi (przykład: obraz „konie”) lub dzielenia jednorodnych, wydłużonych obiektów (przykład: obraz „płuca”). Ujemnym skutkiem takiego działania mogłoby się okazać jednak niepotrzebne dzielenie dużych, niejednorodnych obiektów, których analiza jest podstawą niniejszej pracy.

Podsumowując, skuteczność proponowanego algorytmu zależy od następujących czynników:

- **Wstępnej wielkości segmentów, które mają być łączone.**

Niezależnie od tego, jaki algorytm segmentacji zostanie wybrany, istotne jest takie ustalenie parametrów jego działania, aby uniknąć nadmiernego rozdrobnienia rzeczywistych obiektów na małe segmenty. Często w tym celu stosuje się wstępną procedurę rozmycia obrazu filtrem Gaussa lub filtrami anizotropowymi. Pozwala to na znaczną redukcję liczby segmentów w obrębie poszczególnych obiektów, a co za tym idzie, skraca późniejszy proces ich łączenia. W analizowanych przykładach przyjęto wartość  $\sigma = 1.0$  dla rozmycia Gaussa. Może być ona zwiększona na obrazach o wyraźnie zaznaczonej krawędzi między obiektami (np. obraz „płuca”), jednak w pozostałych przypadkach mogłoby to skutkować powstawaniem zbyt dużych segmentów obejmujących fragmenty kilku rzeczywistych obiektów. Jeśli taka sytuacja będzie miała miejsce, proponowany algorytm łączenia segmentów również może zakończyć się poprawnie, jeśli duże segmenty zostaną rozdzielone linią krawędzi *Canny* (rozwiązanie tego problemu omówiono w rozdziale 6.2 w sekcji *Przygotowanie obrazów do analizy*).

- **Możliwości wyznaczenia przeważającej liczby krawędzi.**

Jeżeli wybrany algorytm detekcji krawędzi nie będzie w stanie wskazać linii rozdzielającej obiekty między sobą lub długość tej linii (krawędzi) będzie zbyt krótka, to analizowane obiekty zostaną połączone w jeden obiekt. W omawianych przykładach zastosowano *algorytm Canny*. Pozwala on na określenie stopnia szczegółowości detekcji krawędzi poprzez zmianę intensywności rozmycia obrazu (parametr *sigma* operatora Gaussa). Dzięki temu można usunąć drobne krawędzie i uzyskać obrys dużych obiektów. Nadmierne zwiększanie tego parametru powoduje jednak usunięcie ważnych, ale mniej wyraźnych krawędzi obrazu, co uniemożliwia uzyskanie poprawnych wyników po etapie łączenia segmentów.

- **Wartości progowej dla kryterium łączenia segmentów.**

Wartość ta decyduje, czy sąsiadujące ze sobą segmenty mogą zostać uznane za część tego samego obiektu. Wpływ na to ma długość granicy między segmentami względem obwodu każdego segmentu z osobna. Dodatkowo brana jest po uwagę obecność i długość linii krawędzi (odpowiednie wzory podano w rozdziale 6.2).

Przedstawiona koncepcja łączenia segmentów z wykorzystaniem informacji o krawędzi jest szczególnie zalecana tam, gdzie znane do tej pory kryteria bazujące na wartościach wszystkich pikseli nie dają dobrego rezultatu. Dotyczy to szczególnie obiektów niejednorodnych, dla których nie można wyznaczyć jednoznacznego kryterium spójności pikseli, ale jest możliwe określenie przybliżonych granic tych obiektów.

### *Detekcja obiektów z użyciem modelu bazującego na kształcie*

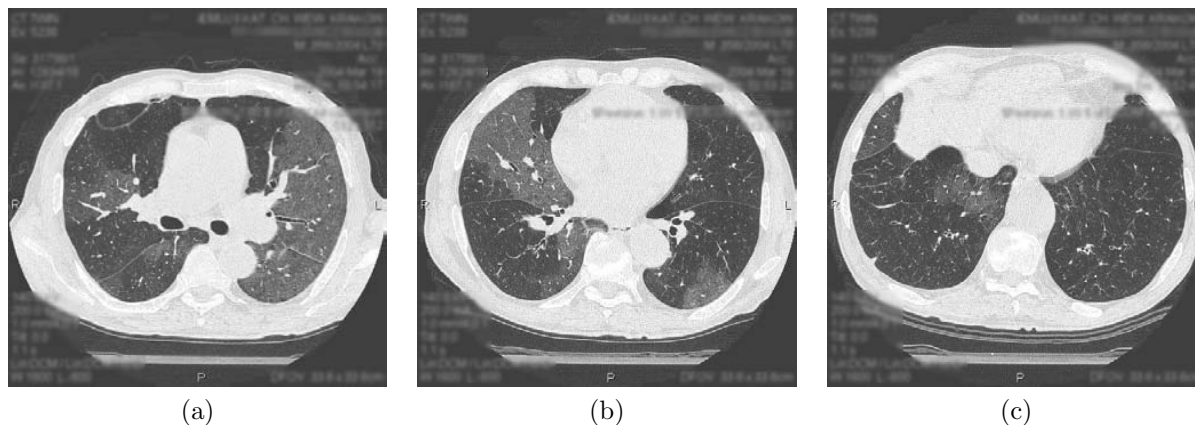
Opisana w poprzednim rozdziale technika segmentacji obrazu z użyciem informacji o krawędzi, pomimo tego, że umożliwia znaczne zmniejszenie liczby segmentów z jakich składają się duże obiekty, to nie jest w stanie wskazać obiektu naszych badań. W tym celu potrzebna jest dodatkowa informacja, która może zostać przekazana w kolejnym kroku procesu analizy obrazu. Informacja ta powinna w jasny sposób precyzować, czym wyróżnia się obiekt będący celem detekcji od innych obiektów i od tła. W sytuacji, gdy analizowany obiekt ma podobny kształt na kolejnych obrazach, można przygotować jego obrys w sposób manualny i wykorzystać go do detekcji obiektów na pozostałych obrazach. Dla przygotowanego w ten sposób modelu obiektu - zwanego również *obiektem wzorcowym*, lub *szablonem obiektu* - można zastosować jeden z wielu algorytmów nakładania obrazów<sup>1</sup> [9, 33, 19, 25] (ang. *Image registration*), tak aby przenieść ten model z jednego obrazu na kolejne (więcej o możliwościach tego typu algorytmów można przeczytać w dodatku B.2 [Algorytmy nakładania obrazów](#), który rozpoczyna się na stronie 124). Technika ta, znana w literaturze jako segmentacja bazująca na atlasie [59, 52, 58, 37] (ang. *Atlas-based image segmentation*), nie jest jednak pozbawiona ograniczeń i niedoskonałości. Może ona być stosowana do analizy serii zdjęć dotyczących konkretnego zagadnienia, a poszukiwany obiekt powinien wykazywać podobieństwo geometryczne dotyczące kształtu, rozmiaru i położenia na obrazie. Tego typu wymagania często są spełnione, kiedy akwizycja obrazu przebiega według ściśle określonych reguł i dotyczy różnych obiektów tego samego typu (np. obszary płuc różnych pacjentów na obrazach tomograficznych klatki piersiowej). Nawet jeśli te wymagania zostaną spełnione, to nie mamy gwarancji pełnej poprawności detekcji. Zazwyczaj w rezultacie otrzymujemy przybliżony kontur obiektu, który nie zawsze pokrywa się z rzeczywistymi granicami obiektu na obrazie. Celem proponowanego w niniejszym rozdziale algorytmu będzie dokładne dopasowanie przygotowanego modelu bazującego na kształcie do granic poszukiwanych obiektów<sup>2</sup>.

---

<sup>1</sup>Użyte sformułowanie *nakładanie obrazów* odnosi się do angielskiego terminu *image registration*. W literaturze polskojęzycznej stosowane są również pojęcia: *rejestracja* lub *pasowanie*.

<sup>2</sup>Użyte w tym wypadku stwierdzenie *dopasowanie modelu* lepiej oddaje specyfikę proponowanego algorytmu aniżeli sformułowanie *nakładanie modelu*, ponieważ w algorytmie uwzględniana jest dodatkowa informacja o strukturze obrazu (szczegółowe informacje podano w rozdziale 7.2).





Rysunek 7.1: Przykładowe zdjęcia tomograficzne klatki piersiowej, które zostaną użyte do omówienia proponowanego w tym rozdziale algorytmu (obrazy pozyskane z *Collegium Medicum Uniwersytetu Jagiellońskiego*).

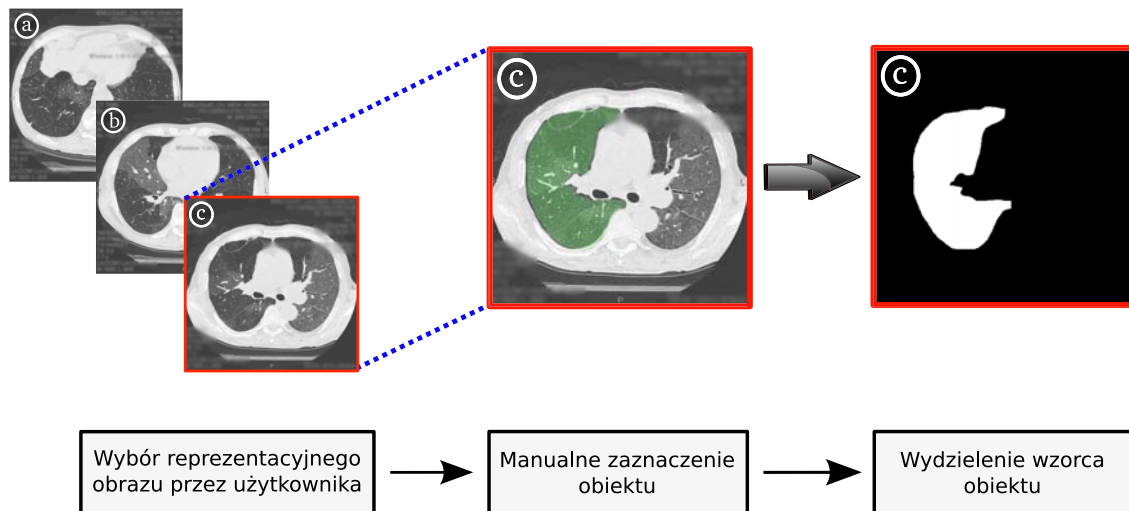
Większość opisanych w kolejnych podrozdziałach algorytmów, jak i proponowana metoda zostaną omówione z użyciem przykładowych skanów tomograficznych płuc (Rys. 7.1). Obraz 7.1a i 7.1b dobrano tak, aby kształt płuc był jak najbardziej zbliżony. Dodatkowo wybrano obraz 7.1c, na którym kształt płuc znacznie różni się od kształtów na pierwszych dwóch obrazach. To zróżnicowanie będzie przydatne, aby omówić możliwości przygotowania ujednoczonego modelu kształtu płuca i problemy jakie z tego wynikają. W kontekście tych problemów zostanie zaprezentowany nowy algorytm dokładnego dopasowania modelu i będą omówione jego możliwości w zakresie poprawy wyników detekcji dla obrazów znacznie różniących się od przyjętego obrazu wzorcowego.

## 7.1 Segmentacja obiektów na obrazie z wykorzystaniem atlasu

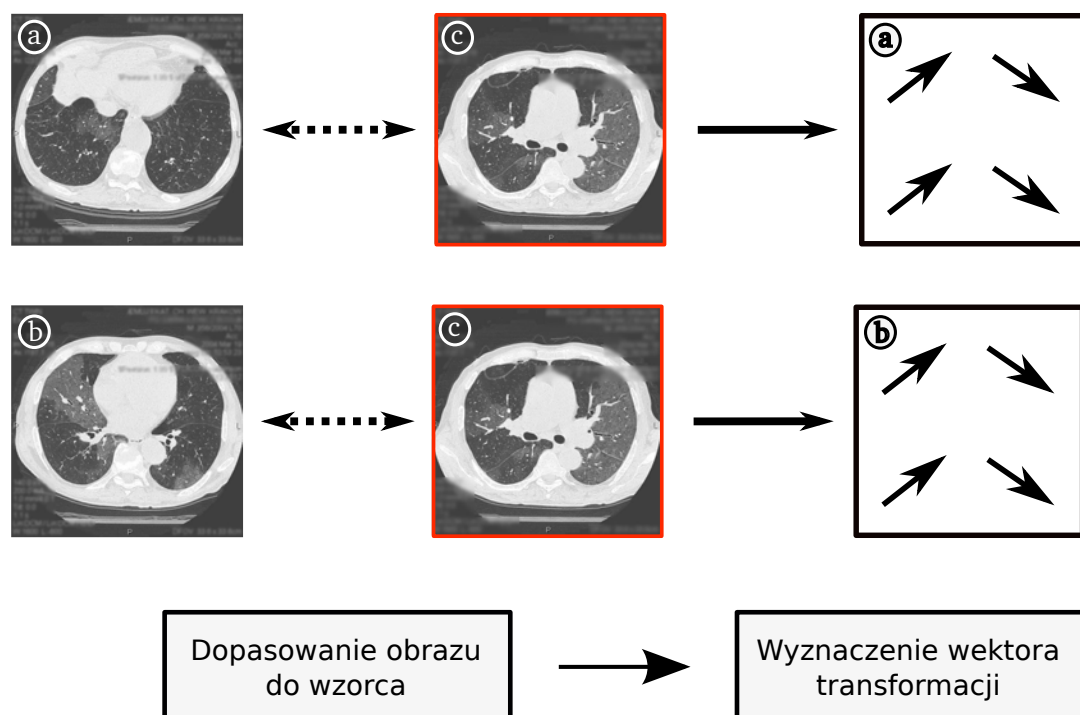
Proponowana w niniejszym rozdziale metoda nie jest w pełni samodzielnym algorytmem detekcji obiektów. Jej celem jest poprawa wyników, jakie możemy uzyskać za pomocą znanych metod segmentacji z wykorzystaniem tzw. atlasu (ang. *Atlas-based image segmentation*). Koncepcja ta zakłada, że wszystkie obrazy do analizy dotyczą tego samego obiektu badań, który został zdeformowany w czasie lub zmieniło się jego położenie<sup>3</sup>. Bazując na tym założeniu wybierany jest wzorcowy obraz i na nim manualnie lub też półautomatycznie zaznaczany jest obiekt badań (Rys. 7.2). Następnie, korzystając z algorytmów nakładania (ang. *Image registration*) poszukiwany jest wektor transformacji

<sup>3</sup>Założenie to nie zawsze jest zgodne z rzeczywistością, ponieważ wzorec opracowany na podstawie wybranego zdjęcia może być stosowany na innych zdjęciach, przedstawiających podobny, lecz nie ten sam obiekt, np. obrys płuc dla jednego pacjenta może być podstawą segmentacji dla innych pacjentów.

jednego obrazu w drugi (Rys. 7.3). Po czym transformacji tej poddawany jest obrys obiektów (Rys. 7.4). W ten sposób można nanieść obrys obiektu na kolejne obrazy, a tym samym zaznaczyć na nich poszukiwany obiekt badań<sup>4</sup>.



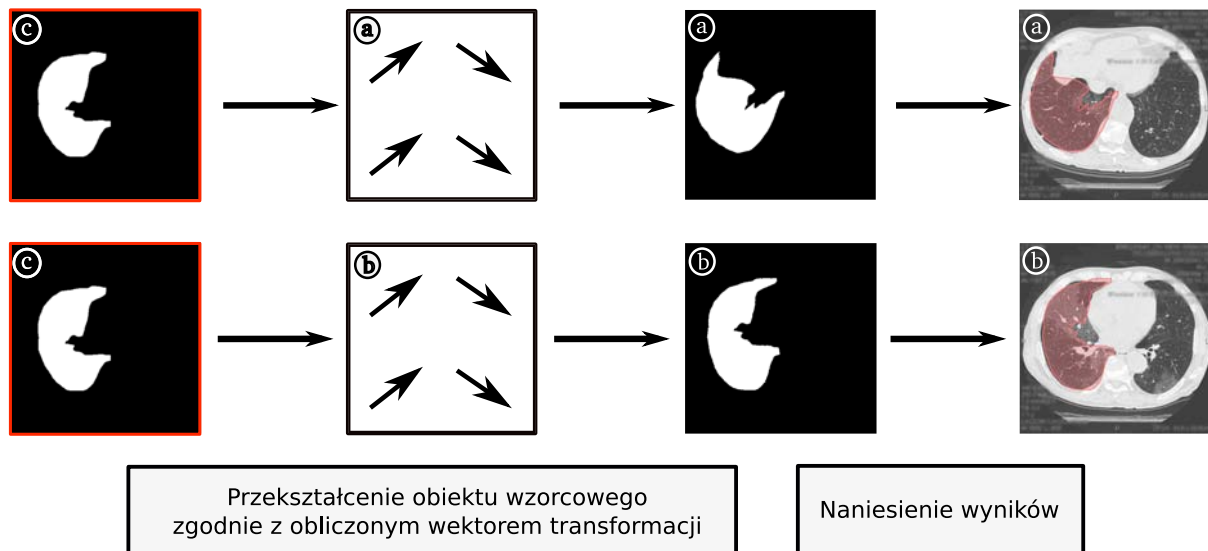
Rysunek 7.2: Pierwszy etap segmentacji bazującej na atlasie - wybór reprezentacyjnego obrazu i przygotowanie wzorca poszukiwanego obiektu (*opracowanie własne*).



Rysunek 7.3: Drugi etap segmentacji bazującej na atlasie - wyznaczenie wektora transformacji obrazu nakładanego (lewa kolumna) do obrazu odniesienia (środkowa kolumna) (*opracowanie własne*).

<sup>4</sup>Widoczne na rysunku 7.4 wyniki dopasowania modelu do obiektu udało się uzyskać z wykorzystaniem narzędzia `bUnwarpJ` [80] opisanego w dodatku B.2. [Algorytmy nakładania obrazów](#).





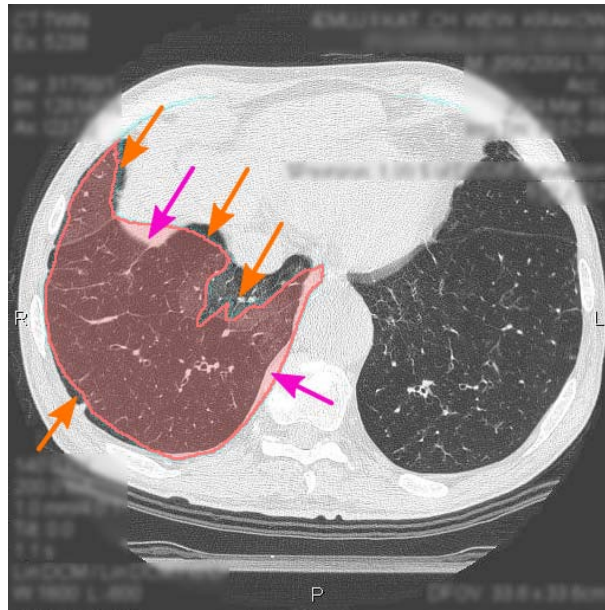
Rysunek 7.4: Trzeci etap segmentacji bazującej na atlasie - deformacja wzorca obiektu zgodnie z obliczonym wektorem transformacji (*opracowanie własne*).

W dalszej części tego rozdziału zostanie omówiona własna metoda umożliwiająca dokładniejsze dopasowanie wzorca do rzeczywistego obiektu oraz zostaną zaprezentowane jej możliwości na kilku przykładowych seriach obrazów. Wcześniej jednak autor zachęca do zapoznania się z dodatkiem B do niniejszej pracy, w którym omówiono podstawy działania algorytmów nakładania obrazów, w celu pełniejszego opisu możliwości i zakresu stosowania proponowanej metody.

## 7.2 Propozycja algorytmu dokładnego dopasowania modelu do obiektu

Opisane w dodatku B wybrane algorytmy nakładania obrazów pokazują, że segmentacja z wykorzystaniem wcześniej przygotowanego modelu (ang. *Atlas-based image segmentation*) nie jest zadaniem łatwym i nie zawsze daje w pełni zadowalający rezultat (Rys. 7.5). Tego typu metody najlepiej sprawdzają się tam, gdzie badany obiekt na poszczególnych zdjęciach zmienia swoją lokalizację, bez znacznych zmian kształtu. W innym przypadku (np. próba dopasowania obrazu 7.1a do obrazu 7.1c) możemy otrzymać tylko przybliżone rozwiązanie, które może być przydatne przy wizualnej ocenie problemu, a nie do wykonania precyzyjnych pomiarów badanych obiektów<sup>5</sup>.

<sup>5</sup>Metody segmentacji z wykorzystaniem atlasu najczęściej używane są na potrzeby wstępnej oceny wizualnej danego problemu. W takich przypadkach brak dokładnego nałożenia obrazu nie musi dyskwalifikować tej metody, ponieważ wyniki za każdym razem oceniane są przez specjalistę z danej dziedziny, np. radiologa.



Rysunek 7.5: Problemy z dokładnością algorytmu nakładania obrazów. Kolorem pomarańczowym zaznaczono fragmenty, które powinny zostać uwzględnione, a kolorem purpurowym te, które powinny zostać pominięte (*opracowanie własne*).

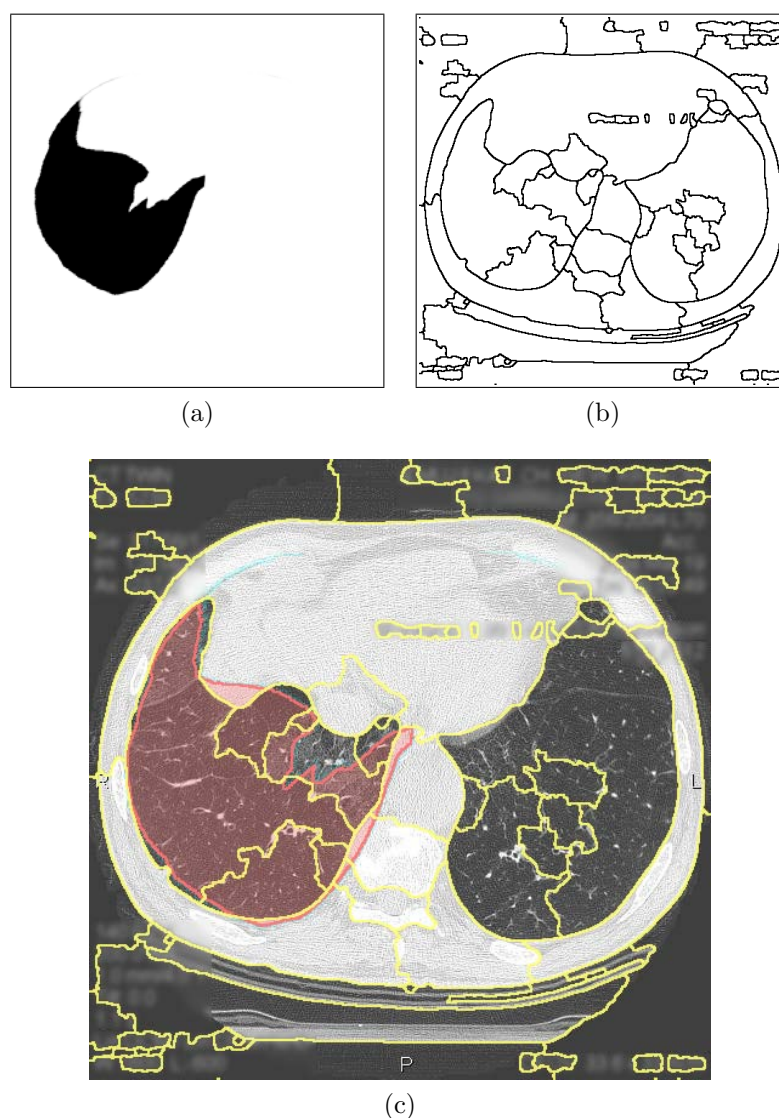
Aby zaradzić tym problemom i poprawić jakość otrzymanych wyników, zostanie zaproponowany algorytm dokładnego dopasowania modelu do obiektu z wykorzystaniem algorytmu działów wodnych. Jego celem będzie taka modyfikacja otrzymanego obiektu (czerwony obszar na rysunku 7.5), aby dołączyć te obszary należące do rzeczywistego obiektu, które nie zostały uwzględnione (zaznaczono na rysunku pomarańczowymi strzałkami) i wykluczyć takie, które wykraczają poza jego granice (zaznaczono purpurowymi strzałkami).

Wśród proponowanych do tej pory w literaturze algorytmów [16, 23], które również stawiają sobie za cel poprawę wyników po etapie nakładania obrazów, warto wskazać:

- Metodę śledzenia poziomicy (ang. *Level set tracking*) stosowaną po etapie globalnego nakładania obrazu bazującą na intensywności (ang. *Intensity-based non-rigid registration*) opisaną w pracy [16]. Pozwala ona na dodatkowy ruch grzbietu obiektu (ang. *Motion of the object interface*) w kierunku wskazanym przez gradient obrazu w celu znalezienia jego rzeczywistych granic.
- Metodę, w której po wstępnym nałożeniu obrazu konstruowany jest znacznik dla algorytmu działów wodnych [23]. Znacznik ten nie jest budowany w oparciu o pełny obiekt, lecz o sam szkielet obiektu. Dzięki temu zalewanie obszarów rozpoczyna się wcześniej i może się zatrzymać przed granicą rzeczywistego obiektu ograniczonego mocną linią gradientu.

## Koncepcja algorytmu

Metody nakładania obrazu w przypadku zmiany kształtu obiektu na kolejnych analizowanych zdjęciach rzadko kiedy pozwalają na dokładne dopasowanie przygotowanego modelu do rzeczywistego obiektu. Problem ten można rozwiązać wprowadzając dodatkowy etap modyfikacji modelu, który zbliży go do rzeczywistych granic poszukiwanego obiektu. Niezbędne w tej sytuacji będzie skorzystanie z dodatkowych informacji, jakie niesie ze sobą analizowany obraz. Tego typu informacją może być wynik wstępnej segmentacji obrazu z wykorzystaniem algorytmu działów wodnych (szerzej opisany w rozdziale 4.1). Algorytm ten, pomimo tego, że często



Rysunek 7.6: Obrazy wejściowe do proponowanego algorytmu dokładnego dopasowania: (a) zdeformowany model uzyskany z modelu wzorcowego po etapie nakładania obrazu, (b) wstępna segmentacja obrazu z użyciem algorytmu działów wodnych, (c) obraz (a) i (b) naniesione na obraz oryginalny 7.1a (opracowanie własne).

proceedzi do nadmiernego rozdrobienia obrazu na małe segmenty dobrze oddaje jego strukturę – linie działów wodnych pokrywają się z granicami rzeczywistych obiektów. Ilością, a zarazem wielkością segmentów można sterować wprowadzając dodatkowy etap rozmycia obrazu, co wpływa na liczbę lokalnych minimów, które stanowią punkty startowe dla algorytmu zalewania. Nakładając na obraz odniesienia (Rys. 7.1a) obiekt wzorcowy po deformacji (Rys. 7.6a) oraz linie działów wodnych (Rys. 7.6b) widać wyraźnie, że segmenty działów wodnych dobrze wpasowują się w docelowy obiekt, podczas gdy model po deformacji nie styka się bezpośrednio z jego granicą. Bazując na tym spostrzeżeniu, można dodać do modelu punkty tych segmentów, których liczba pikseli zawartych w obrębie modelu jest większa od liczby pikseli leżących poza nim. W przeciwnym wypadku, gdy większość punktów danego segmentu leży poza obrębem modelu, można przyjąć, iż to model wykracza poza granice rzeczywistego obiektu i należy od niego odjąć punkty wspólne z analizowanym segmentem.

W proponowanym algorytmie nie zakłada się jednak dalszej modyfikacji zdeformowanego modelu, tylko budowę nowego modelu w oparciu o segmenty działów wodnych i informację o położeniu uprzednio otrzymanego modelu. Korzystając z algorytmu łączenia regionów (ang. *Region merging*) będą dodawane do siebie te segmenty, których przeważająca liczba punktów znajduje się w obrębie zdeformowanego modelu. Otrzymane w ten sposób wyniki będą identyczne z wynikami otrzymanymi przy powyżej przedstawionym podejściu modyfikacji modelu, polegającym na dodawaniu i odejmowaniu segmentów leżących na jego granicach.

## Realizacja algorytmu

Używany w tej metodzie algorytm łączenia segmentów będzie miał prostszy schemat działania, niż algorytm zaproponowany wcześniej w rozdziale 6.2. Tam zdecydowano się na reprezentację segmentów za pomocą grafów, a sam proces łączenia przebiegał w ściśle określonej kolejności, zależnej od wartości kryterium i grupy, do jakiej dany segment został zaklasyfikowany. Tutaj natomiast, po wstępnym etapie oceny segmentów wyznaczonych przez działy wodne, podejmowana jest decyzja, czy dany segment będzie wchodził w skład nowo tworzonego modelu, czy też nie. Kolejne kroki algorytmu są następujące:

1. Wygładź obraz wejściowy  $IM_{in}$  filtrem Gaussa i przeprowadź segmentację obrazu po rozmyciu  $IM_{blur}$  za pomocą algorytmu działów wodnych. Parametr  $\sigma$  rozmycia Gaussa ustal tak, aby poszczególne obiekty w trakcie segmentacji nie zostały połączone w jeden segment (szerzej opisane w dalszej części rozdziału, w punkcie 7.2). W wyniku powstanie poindeksowany obraz segmentów  $IM_{us}$ .

2. Znajdź macierz transformacji  $TM$  obrazu wzorcowego  $IM_{tp}$  do aktualnie analizowanego obrazu wejściowego  $IM_{in}$  (szerzej opisano w dodatku B).
3. Zastosuj transformację do wzorcowego modelu obiektu  $OM_{tp}$ . Otrzymamy w ten sposób zdeformowany model  $OM_{in}$  aktualnie analizowanego obrazu (szerzej opisano w rozdziale 7.1).
4. Zlicz liczbę segmentów na obrazie  $IM_{ws}$  i zapisz do zmiennej *number\_of\_labels*.
5. Przygotuj wektory *label\_inside* i *label\_outside* o długości *number\_of\_labels*. Będą one służyć do zliczania liczby pikseli każdego segmentu leżących w obrębie modelu  $OM_{in}$  lub poza nim.
6. Dla każdego piksela obrazu  $IM_{ws}$  sprawdź, czy piksel o tych samych współrzędnych na obrazie  $OM_{in}$  leży w obrębie modelu, czy poza nim. Jeśli wewnątrz, to zwiększ odpowiednią pozycję wektora *label\_inside*, w przeciwnym wypadku zwiększ *label\_outside*.
7. Utwórz pusty obraz  $OM_{out}$ , który będzie zawierał wynikowy model obiektu. Dla każdego segmentu z obrazu  $IM_{ws}$  sprawdź odpowiednie wartości wektorów *label\_inside* i *label\_outside*. Jeśli dla segmentu o numerze *current\_index* prawdziwe jest wyrażenie

$$label\_inside[current\_index] > label\_outside[current\_index],$$

to dodaj wszystkie piksele tego segmentu do modelu  $OM_{out}$ .

W przedstawionym algorytmie można wyróżnić dwie zasadnicze części. Pierwsza z nich (punkty 1-3) nie stanowi właściwej części proponowanego algorytmu dokładnego dopasowania, tylko pokazuje jego kontekst użycia i sposób przygotowania danych wejściowych. W punkcie (1) wykonywana jest wstępna segmentacja obrazu  $IM_{in}$  z użyciem algorytmu działów wodnych. W punkcie (2) i (3) nakładany jest wzorcowy model  $OM_{tp}$  na aktualnie analizowany obraz odniesienia  $IM_{in}$ . Otrzymany w ten sposób zdeformowany model  $OM_{in}$  w drugiej części algorytmu będzie podstawą oceny poszczególnych segmentów. W punkcie (6) sprawdzane są poszczególne punkty każdego segmentu, czy leżą w obrębie modelu  $OM_{in}$ , czy poza nim. Ostatecznie w punkcie (7) tworzony jest nowy model  $OM_{out}$ , który składa się wyłącznie z segmentów, spełniających postawiony tam warunek.

Dodatkowo, w celu ułatwienia implementacji schematu dla właściwej części algorytmu (tj. punktów 4-7) pokazano jego ogólny zapis w postaci tzw. *pseudo-kodu*.

```

number_of_labels ← get_max_label_index(IMws)
declare vector label_inside of size number_of_labels
declare vector label_outside of size number_of_labels
for y = 1 to image_width do
  for x = 1 to image_height do
    current_index ← IMws[x, y]
    if OMin[x, y] = 1 then
      label_inside[current_index] ++
    else
      label_outside[current_index] ++
    end if
  end for
end for
for y = 1 to image_width do
  for x = 1 to image_height do
    current_index ← IMws[x, y]
    if label_inside[current_index] > label_outside[current_index] then
      OMout[x, y] ← 1
    else
      OMout[x, y] ← 0
    end if
  end for
end for

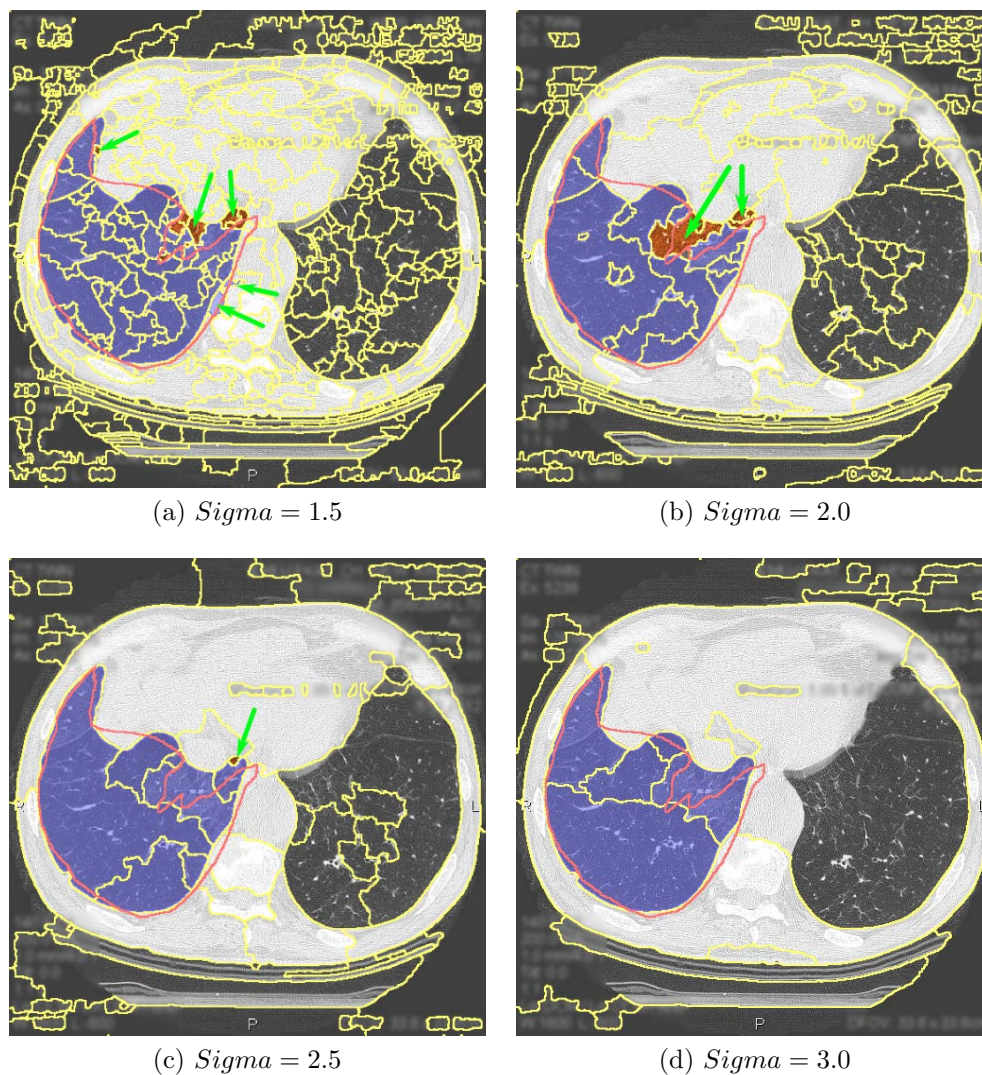
```

Wyżej przedstawiony algorytm może być łatwo zaimplementowany przy użyciu dowolnego języka programowania. Zaproponowano w nim użycie jedynie prostych struktur programowych, takich jak *wektory* (tablice jednowymiarowe) i *macierze* (tablice dwuwymiarowe), co pozwala na skorzystanie również z prostych języków strukturalnych, np. *C* lub *Pascal*. Algorytm ten jest również bardzo szybki w działaniu, ponieważ wymaga jedynie dwukrotnego przejścia po wszystkich punktach obrazu, a dla każdego z nich wykonywane są jedynie elementarne operacje takie jak przypisanie i porównanie wartości.



## Sterowanie dokładnością działania algorytmu

Jak wynika z opisanego w poprzedniej części schematu algorytmu, jedynym parametrem sterującym jego działaniem jest wartość  $\sigma$  rozmycia Gaussa. Parametr ten w decydujący sposób wpływa na wielkość segmentów otrzymywanych algorytmem działów wodnych, a co za tym idzie będzie miał również istotny wpływ na wynik działania zaproponowanego algorytmu łączenia segmentów. Na rysunku 7.7 przedstawiono wyniki działania algorytmu dla różnych wartości  $\sigma$ : rysunek 7.7a - 1.5, rysunek 7.7b - 2.0, rysunek 7.7c - 2.5, rysunek 7.7d - 3.0. Dla niższej wartości parametru (dwa



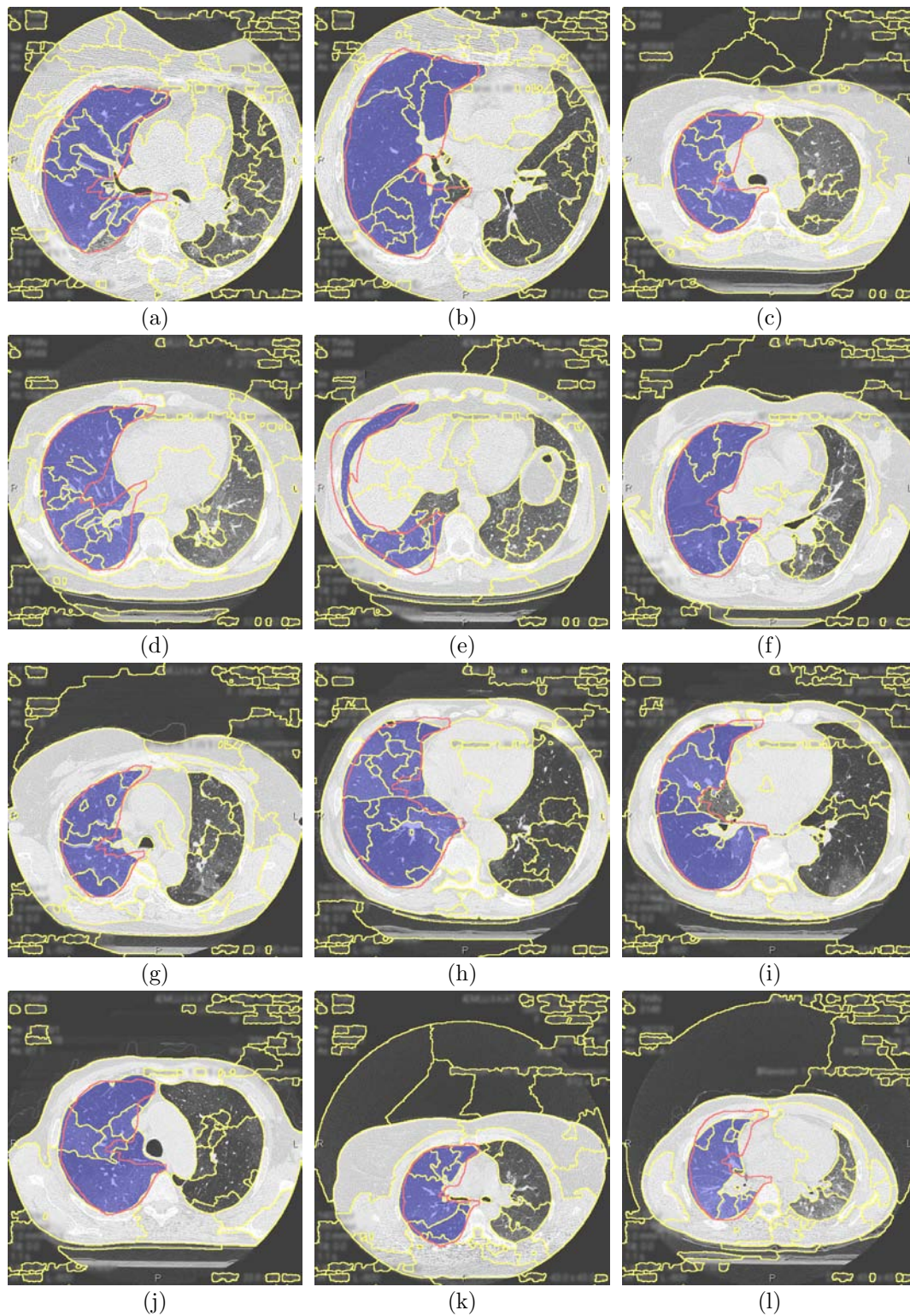
Rysunek 7.7: Wyniki działania proponowanego algorytmu dokładnego dopasowania dla różnych wartości  $\sigma$  rozmycia Gaussa. Oznaczenia: linia żółta - działy wodne; linia czerwona - wzorcowy obiekt po deformacji; obszar niebieski - wyniki proponowanego algorytmu dokładnego dopasowania. Zielone strzałki wskazują na segmenty, które zostały pominięte (obszar pomarańczowy) lub też błędnie wliczone w obszar poszukiwanego obiektu (*opracowanie własne*).

pierwsze obrazy) pomimo widocznej poprawy wyniku końcowego względem wcześniej wyznaczonego modelu, widoczne są miejsca (zaznaczono zielonymi strzałkami), gdzie wynik nie jest zgodny z granicą rzeczywistego obiektu. Natomiast na dwóch kolejnych obrazach wskazane miejsca zostały zdecydowanie lepiej sklasyfikowane – na obrazie 7.7d poprawnie połączono wszystkie segmenty, a na obrazie 7.7c został pominięty jeden mały segment. Pomimo tego, że najlepsze rezultaty osiągnięto dla parametru  $\sigma = 3.0$ , to warto zwrócić uwagę, że taka wysoka wartość tego parametru nie pozwala na wydzielenie obszaru lewego płuca (na obrazie jest to płuco po prawej stronie) i na jego dalszą detekcję przy użyciu proponowanej metody.

Podsumowując zebrane spostrzeżenia można stwierdzić, że dokładność proponowanego algorytmu zależy przede wszystkim od odpowiedniego doboru parametru  $\sigma$  funkcji Gaussa. Dla bardzo małej wartości tego parametru lub jej zerowej wartości (brak wygładzania obrazu) algorytm wyznaczy obszar zbliżony do zdeformowanego modelu obiektu uzyskanego po etapie nakładania obrazów. Z drugiej strony, przy zbyt dużej wartości może nastąpić pominięcie poszukiwanego obiektu na etapie wstępnej segmentacji algorytmem działów wodnych. Zaleca się zatem tak dobierać wartość parametru  $\sigma$ , aby poszukiwany obiekt został podzielony na kilka części. W następnym kroku algorytm podejmie próbę ich połączenia korzystając z informacji o położeniu wcześniej wyznaczonego modelu obiektu.

Opisane powyżej kroki postępowania przetestowano również na pozostałych skanach tomograficznych płuc, do których autor miał dostęp dzięki uprzejmości lekarzy z II Katedry Chorób Wewnętrznych Collegium Medicum Uniwersytetu Jagiellońskiego. Na rysunku 7.8 pokazano wyniki działania proponowanego algorytmu na kolejnych 12 obrazach, które wybrano z serii 25 obrazów tak, aby były one jak najbardziej zróżnicowane pod względem kształtu płuc. Warto zauważyć, że już sam etap deformacji wzorcowego obiektu pozwala na dobre wskazanie obszaru prawego płuca na kolejnych obrazach. Niestety jest to detekcja przybliżona, która nie pozwala na dokładną detekcję brzegu obiektu. Zastosowanie wprowadzonej w tym rozdziale metody umożliwia lepsze wpasowanie modelu w rzeczywisty obiekt podnosząc znacznie precyzję końcowej detekcji.

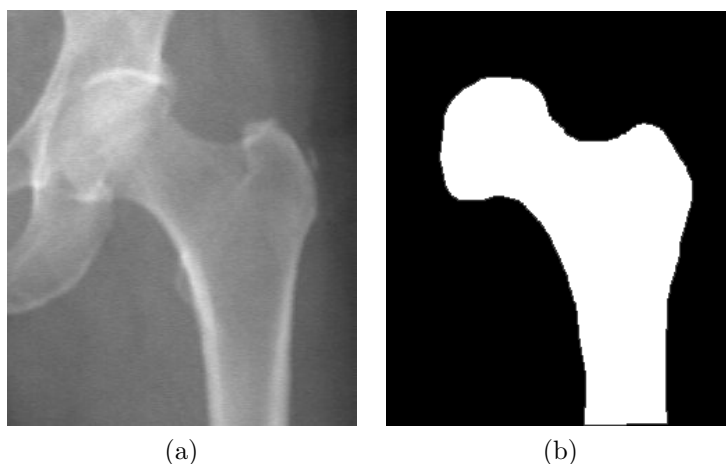




Rysunek 7.8: Wynik działania proponowanego algorytmu dokładnego dopasowania dla zdjęć tomograficznych klatki piersiowej. Oznaczenia: linia żółta - działy wodne; linia czerwona - wzorcowy obiekt po deformacji; obszar niebieski - wyniki proponowanego algorytmu dokładnego dopasowania (*opracowanie własne*).

## Testy algorytmu dla obrazów kości udowej

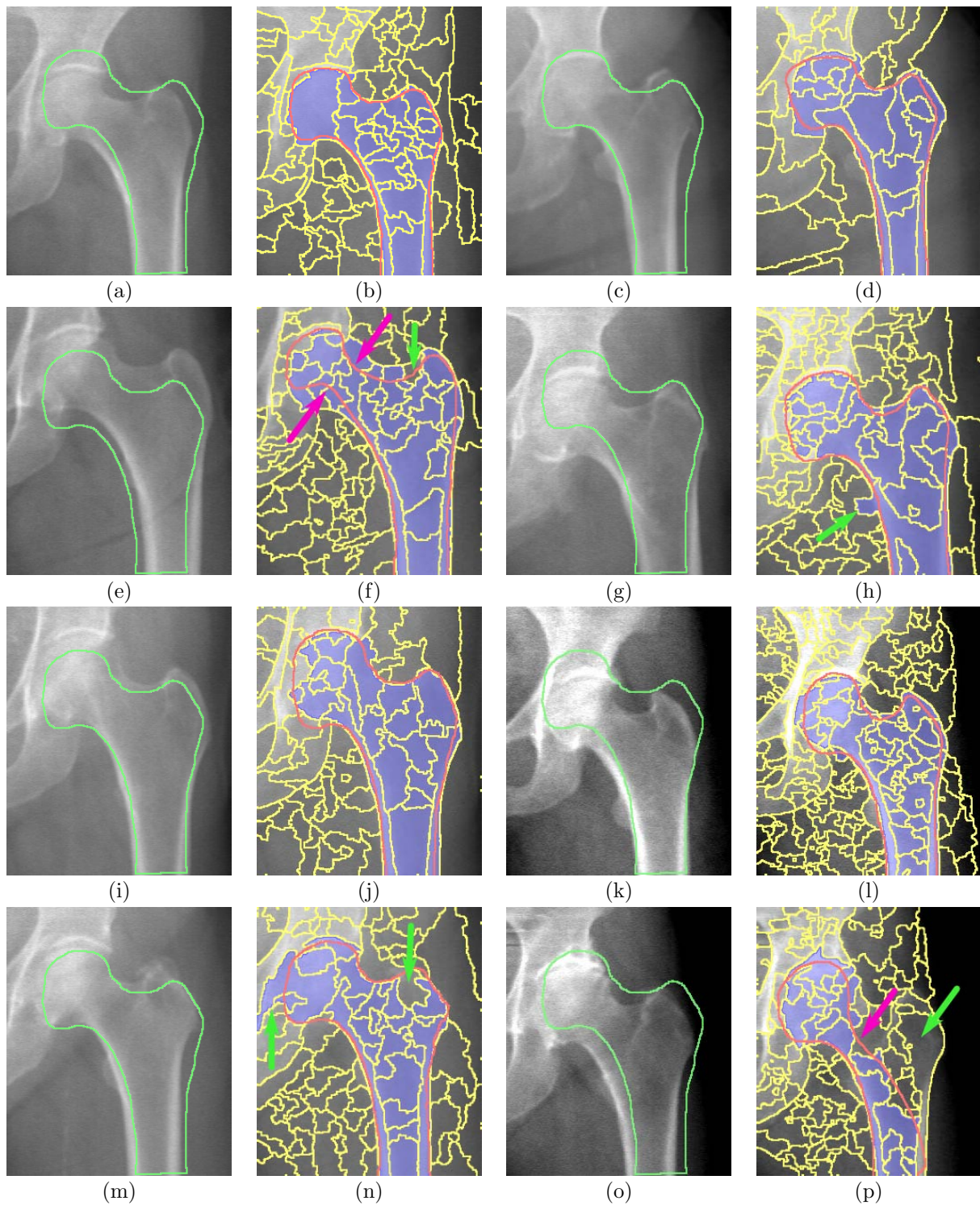
Do tej pory algorytm był szczegółowo omawiany z wykorzystaniem obrazów tomograficznych klatki piersiowej, gdzie obiektem analizy było prawe płuco. Tutaj zostaną również zaprezentowane wyniki jego działania dla obrazów radiologicznych kości udowej, które były podstawą analizy algorytmu opisanego w rozdziale 6.2. W odróżnieniu od obszaru płuc, który był wyraźnie ciemniejszy od korpusu ciała, obszar kości udowej tylko nieznacznie różni się od otoczenia, a jego granica w wielu miejscach jest rozmyta i płynnie łączy ze sobą poszczególne obiekty. Ma on również niejednorodną jasność pikseli, co skutecznie uniemożliwia stosowanie metod opartych na binaryzacji obrazu. Jednak sama kość udowa na poszczególnych obrazach, które były analizowane, nie zmienia tak bardzo swojego kształtu, jak to miało miejsce w przypadku obszaru prawego płuca. Fakt ten przemawia za użyciem metody nakładania obrazu do detekcji poszukiwanego obiektu.



Rysunek 7.9: Wzorcowy obraz kości udowej (a) i manualnie przygotowany jej obrys (b) (*opracowanie własne*).

Postępując zgodnie ze schematem opisanym w rozdziale 7.1, z kilku obrazów wybrano jeden i na nim zaznaczono kontur kości udowej (Rys. 7.9). Informacje te stanowiły podstawę do przygotowania zdeformowanego modelu obiektu dla pozostałych analizowanych obrazów. Wyniki tej operacji naniesiono w postaci czerwonej linii na rysunku 7.10 (obrazy w drugiej i czwartej kolumnie). Dalej przeprowadzono wstępną segmentację obrazu z wykorzystaniem algorytmu działów wodnych (parametr  $\sigma$  rozmycia obrazu funkcją Gaussa o wartość 1.0). Linie działów wodnych zaznaczono kolorem żółtym na obrazach w kolumnie drugiej i czwartej na rysunku 7.10. Ostatecznie uruchomiono zaproponowany algorytm dokładnego dopasowania i przygotowano wynikowy obszar obiektu (zaznaczono kolorem niebieskim).

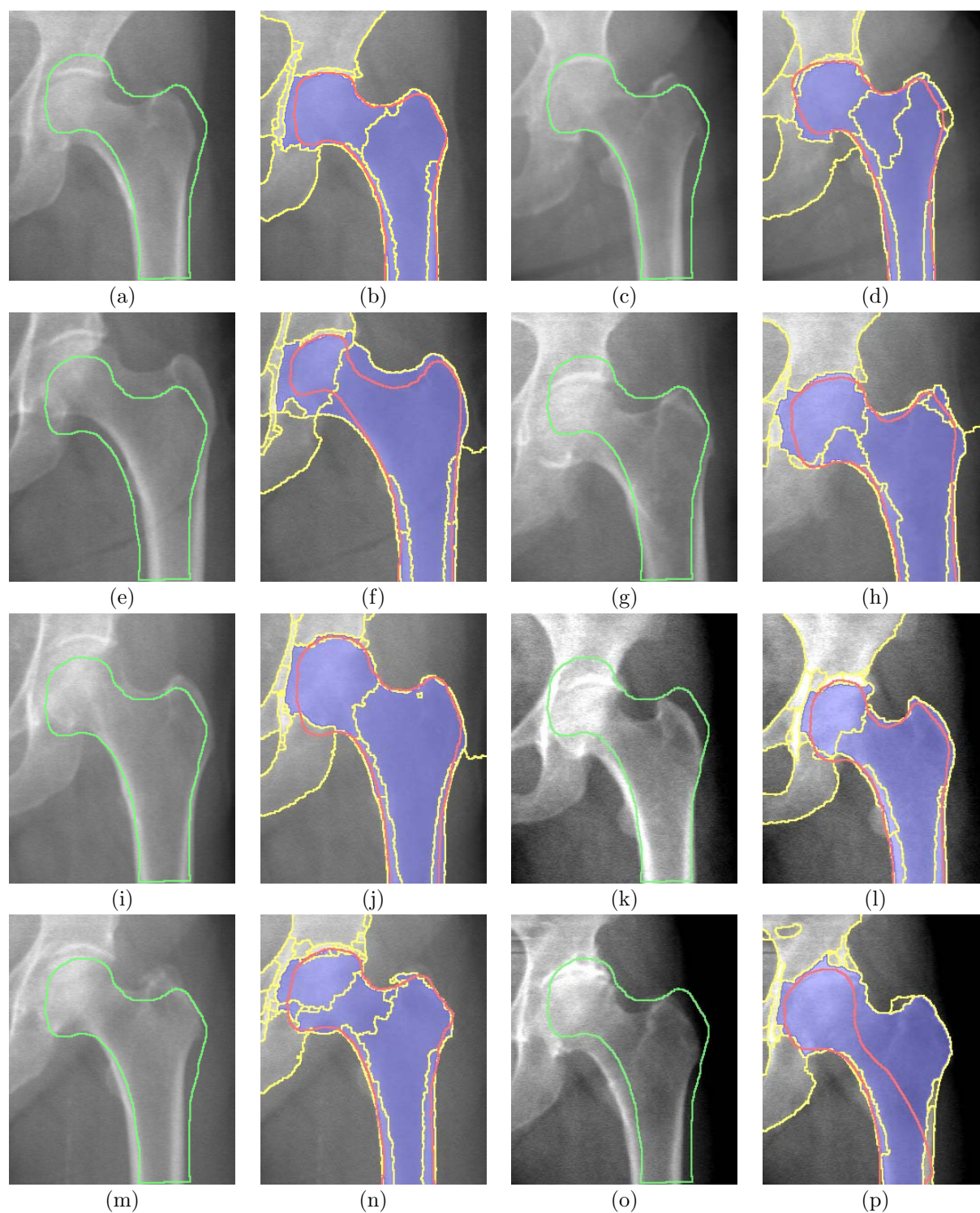




Rysunek 7.10: Prezentacja działania proponowanego algorytmu dokładnego dopasowania dla obrazów kości udowej. W pierwszej i trzeciej kolumnie naniesiono na analizowane obrazy obrys wzorcowego obiektu. W drugiej i czwartej kolumnie naniesiono na obraz oryginalny linie działów wodnych (kolor żółty), zdeformowany model (kolor czerwony) oraz wynik działania proponowanego algorytmu (kolor niebieski). Strzałkami oznaczono błędy w detekcji (*opis w tekście, opracowanie własne*).

Analizując wyniki przedstawione na rysunku 7.10 warto zauważyć, że zdeformowany model przygotowany przy użyciu algorytmu nakładania obrazów na większości obrazów dosyć dokładnie pokrywa się z rzeczywistymi granicami kości udowej. Wyjątkiem są jednak obrazy 7.10f i 7.10p, gdzie to pokrycie jest niedokładne w niektórych obszarach kości (zaznaczono purpurową strzałką). Zastosowanie zaproponowanego algorytmu dokładnego dopasowania przynosi jednak poprawę tylko dla obrazu 7.10f, podczas gdy na obrazie 7.10p nie daje oczekiwanych rezultatów. Dodatkowo można zauważyć błędną detekcję na innych obrazach, w tym 7.10h i 7.10n (zaznaczono zielonymi strzałkami). Prezentowane błędy w działaniu zaproponowanej metody są skutkiem dużego rozdrobienia segmentów – dotyczy to głównie przypadku 7.10p oraz niedokładnej detekcji granic – dotyczy to przypadku 7.10h i 7.10n. Przeprowadzone próby zwiększenia wielkości segmentów poprzez zwiększenie parametru  $\sigma$  rozmycia obrazu filtrem Gaussa przed wykonaniem segmentacji działaniami wodnymi nie dawało lepszych rezultatów. Skutkowało to natomiast tworzeniem niepoprawnych połączeń segmentów, które nie były zgodne ze strukturą analizowanych obrazów. Widoczne to było głównie w procesie błędnego tworzenia segmentów zawierających zarówno fragmenty kości jak i tła.

Aby wyeliminować opisane problemy wstępnej segmentacji obrazu przeprowadzonej algorytmem działów wodnych użyto algorytmu opisanego w rozdziale 6.2. Jego głównym celem jest zwiększenie rozmiaru segmentów oparte na informacji pozyskanej z algorytmu detekcji krawędzi *Canny*. Dodatkowo, dzięki wstępnej obróbce obrazu przeprowadzanej na początkowym etapie użycia tego algorytmu (szerzej opisano w rozdziale 6.2) możliwe jest rozdzielenie tych segmentów, które w całości są przecięte linią krawędzi (może to być przydatne dla przypadku błędnej detekcji na obrazie 7.11h). Rezultaty przedstawione na rysunku 7.11 pokazują, że zastosowanie tego algorytmu pozwala na poprawną detekcję i eliminuje wskazane uprzednio błędy w detekcji – obraz 7.11h, 7.10n, 7.11p. Fakt ten może być kolejnym dowodem skuteczności zaproponowanego algorytmu łączenia segmentów.



Rysunek 7.11: Prezentacja możliwości poprawy błędów w detekcji kości udowej z rysunku 7.10 po zastosowaniu zaproponowanego w rozdziale 6.2 algorytmu łączenia segmentów (*opracowanie własne*).

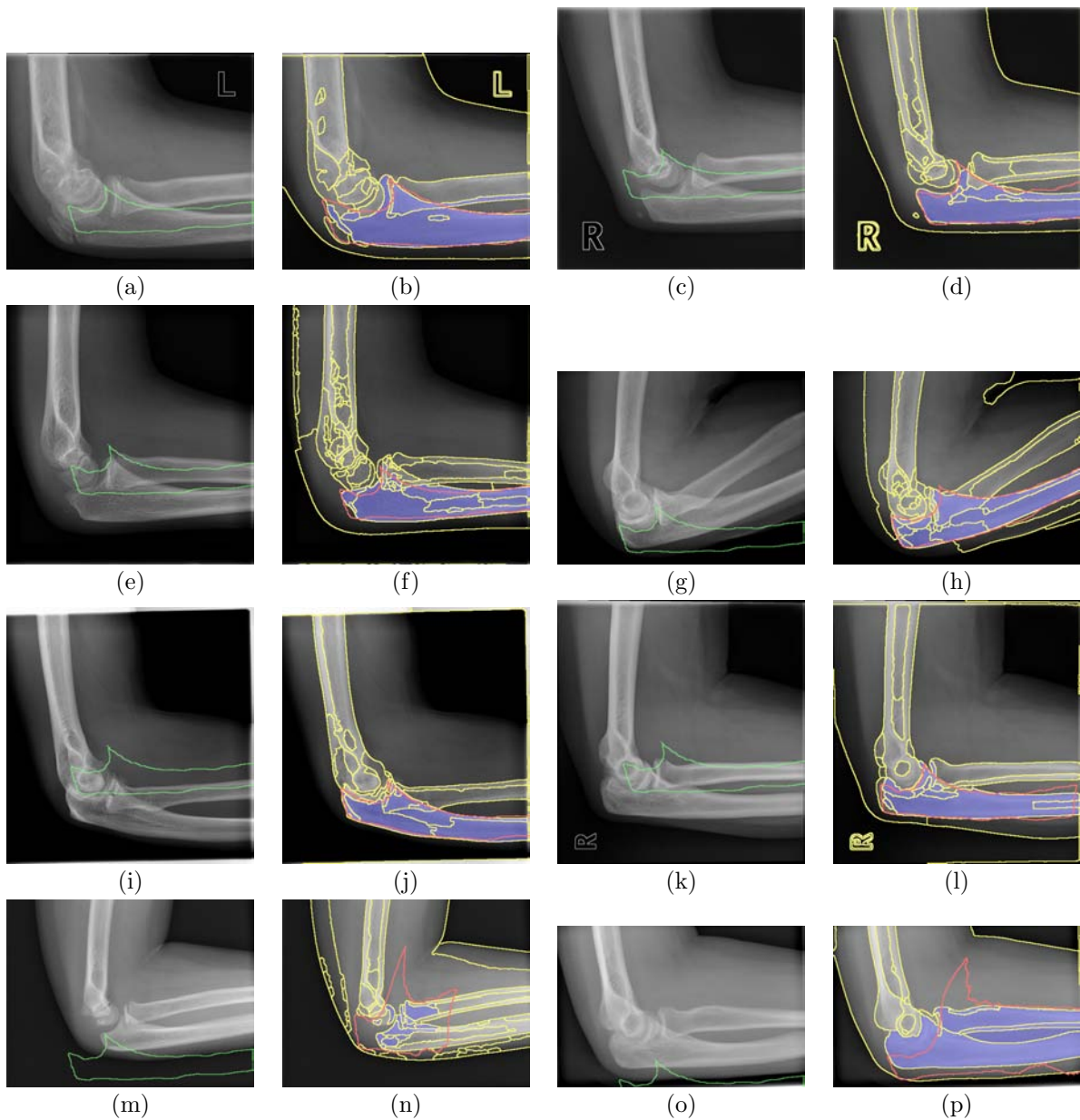


## Dodatkowe testy algorytmu

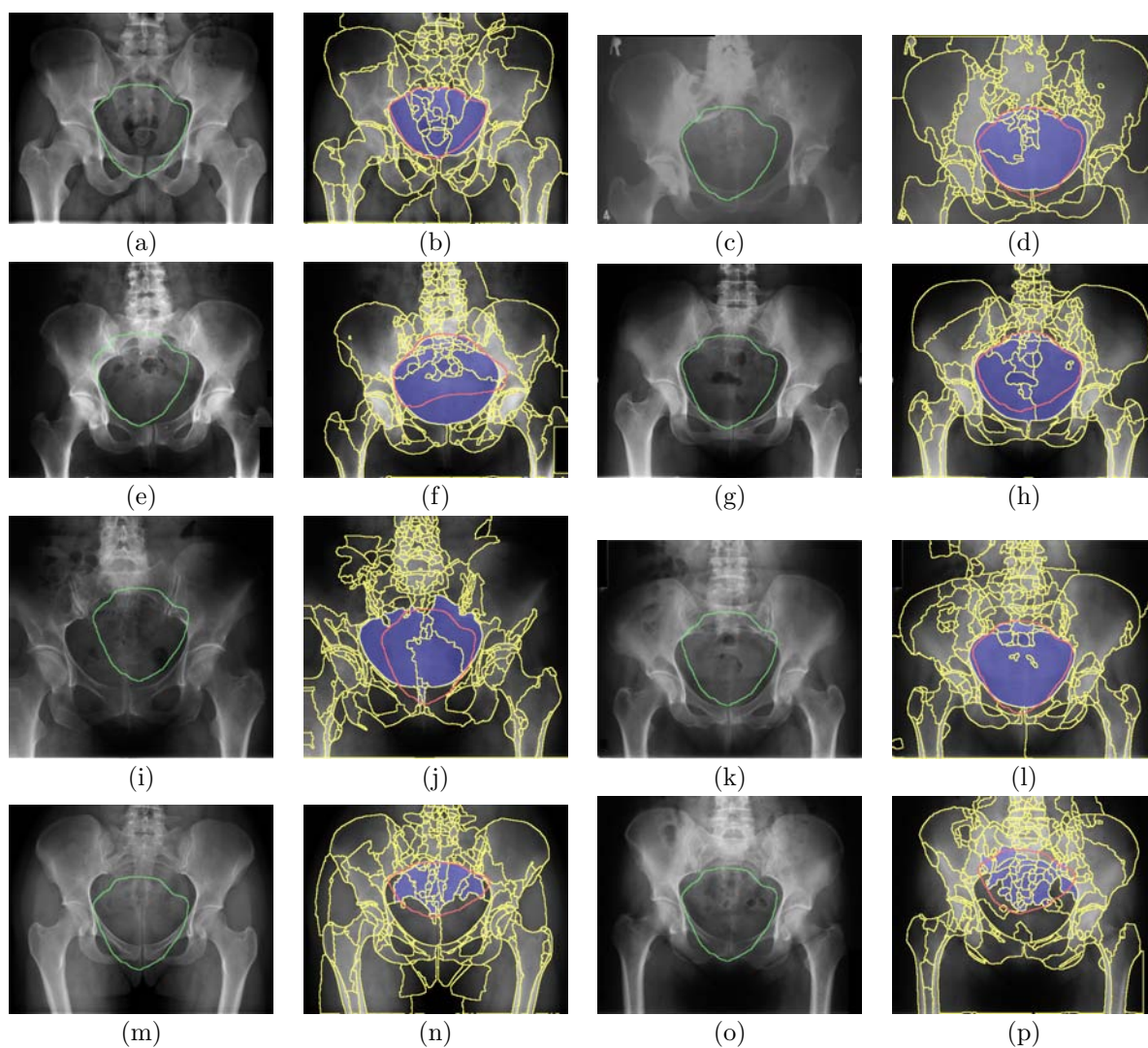
Opisane powyżej zalety łączenia obu przygotowanych algorytmów (Rozdz. 6.2 i 7.2) na przykładzie obrazów radiologicznych kości udowej, zachęciły autora do przeprowadzenia dodatkowych testów na innych obrazach. W tym celu zostały podjęte starania o uzyskanie dostępu do zbioru obrazów medycznych gromadzonych w ramach projektu *Image Retrieval in Medical Applications - IRMA 2009*<sup>6</sup>. Po uzyskaniu licencji na przetwarzanie kilku serii obrazów radiologicznych na potrzeby pracy naukowej przystąpiono do ich analizy. Najpierw dla każdej serii zdjęć wybrano jedno, na którym zaznaczono jeden obiekt. Następnie obiekt ten z wykorzystaniem technik nakładania obrazu był przenoszony na kolejne obrazy. Ostatecznie wykonano wstępną segmentację obrazu i uruchomiono proponowany algorytm ścisłego dopasowania modelu do obiektu. Niestety, podobnie jak to miało miejsce w przypadku kości udowej, tak i tutaj wyniki, które uzyskano dla każdej serii obrazów nie były akceptowalne. Podjęta w kolejnym kroku próba redukcji liczby segmentów z wykorzystaniem własnego algorytmu łączenia segmentu w każdym wypadku umożliwiła poprawę wyników. Wyniki, które w ten sposób udało się uzyskać, zostały przedstawione odpowiednio na rysunkach: kość łokciowa (7.12), jama miednicy (7.13), druga kość śródreżcza lewej ręki (7.14), kość piszczelowa (7.15), kość udowa (7.16), kość piętowa (7.17), kość ramienna (7.18). W celu prezentacji wyników wybrano sześć przykładowych obrazków dla każdego typu obiektu, dla których przedstawiony tok analizy zakończył się sukcesem oraz dwa obrazy, na których nie udało się przeprowadzić poprawnej detekcji wybranego obiektu. Każdorazowo problemy te wynikały z dużej niedokładności etapu pasowania obrazów, uniemożliwiając tym samym podjęcie skutecznej próby ścisłego dopasowania modelu do obiektu z wykorzystaniem przygotowanego algorytmu. Pomimo pojawiających się błędów rezultaty, które udało się uzyskać na większości zdjęć wydają się na tyle zachęcające, że metoda ta może być z powodzeniem używana do szybkiej detekcji wybranych obiektów na obrazie.

---

<sup>6</sup>Opis tego zbioru danych został przedstawiony w rozdziale 1. [Wstęp](#).

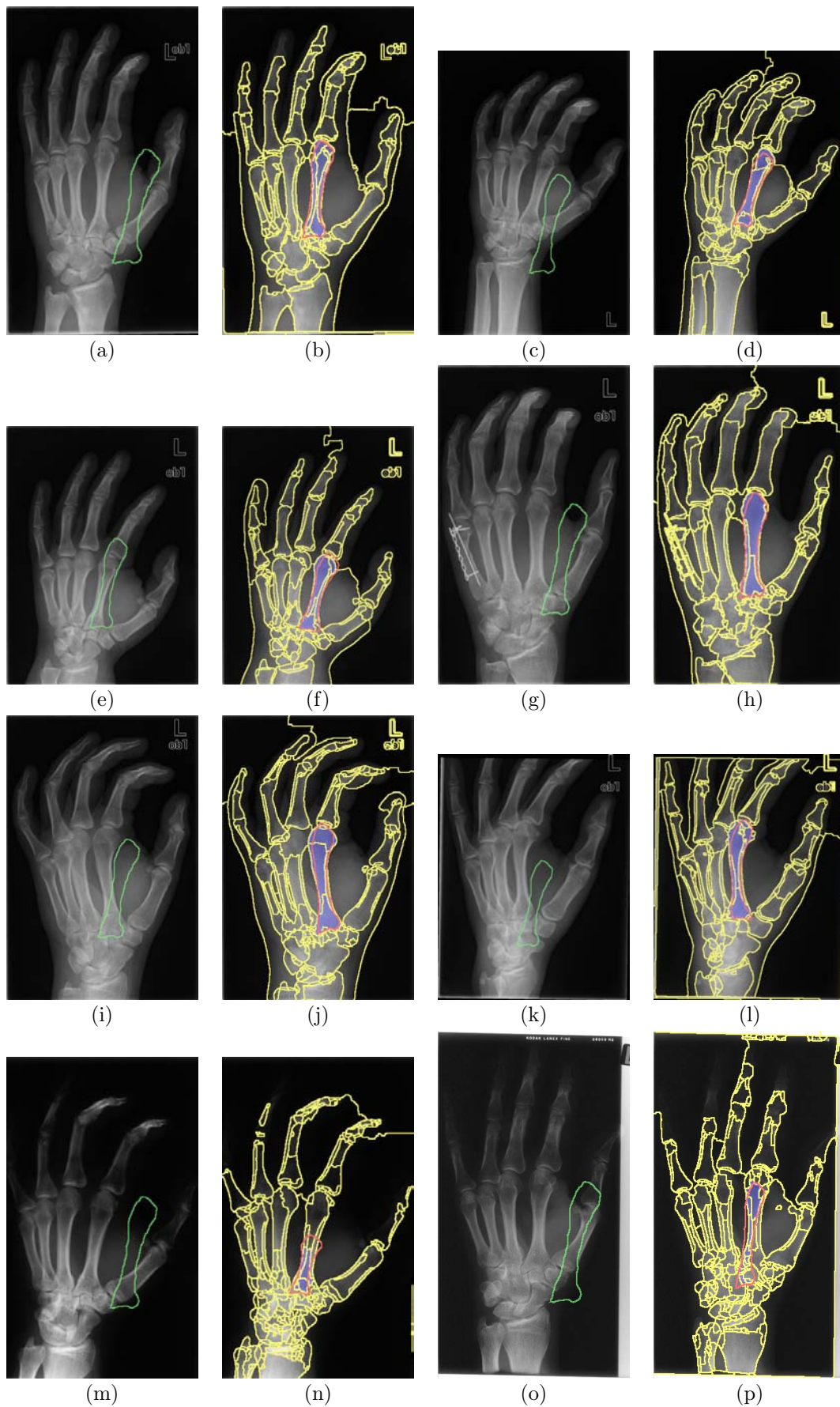


Rysunek 7.12: Wyniki detekcji kości łokciowej z wykorzystaniem zaproponowanych w niniejszej pracy algorytmów łączenia segmentów (Rozdz. 6.2) i ścisłego dopasowania modelu do obiektu (Rozdz. 7.2) (*opracowanie własne*).

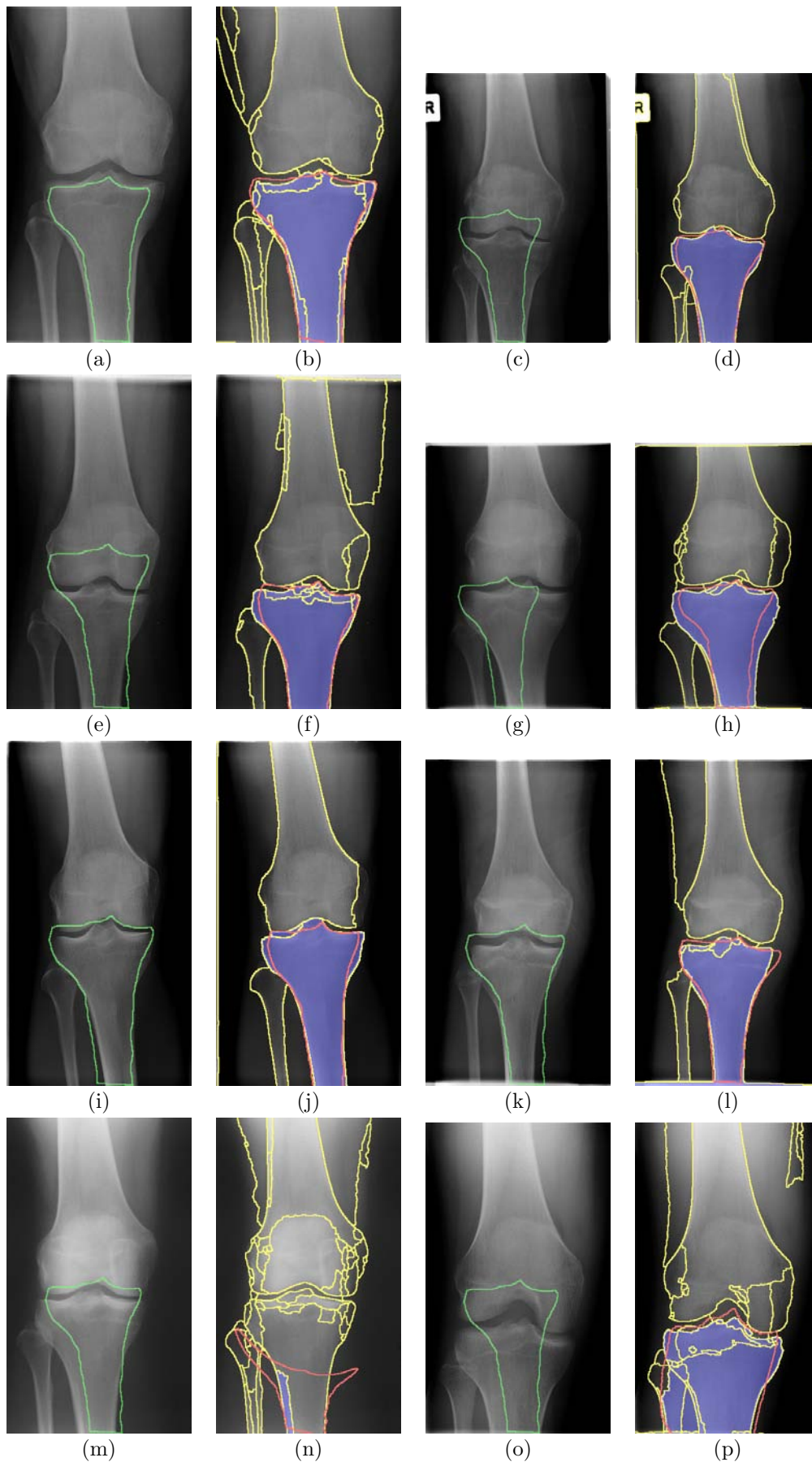


Rysunek 7.13: Wyniki detekcji jamy miednicy z wykorzystaniem zaproponowanych w niniejszej pracy algorytmów łączenia segmentów (Rozdz. 6.2) i ścisłego dopasowania modelu do obiektu (Rozdz. 7.2) (*opracowanie własne*).

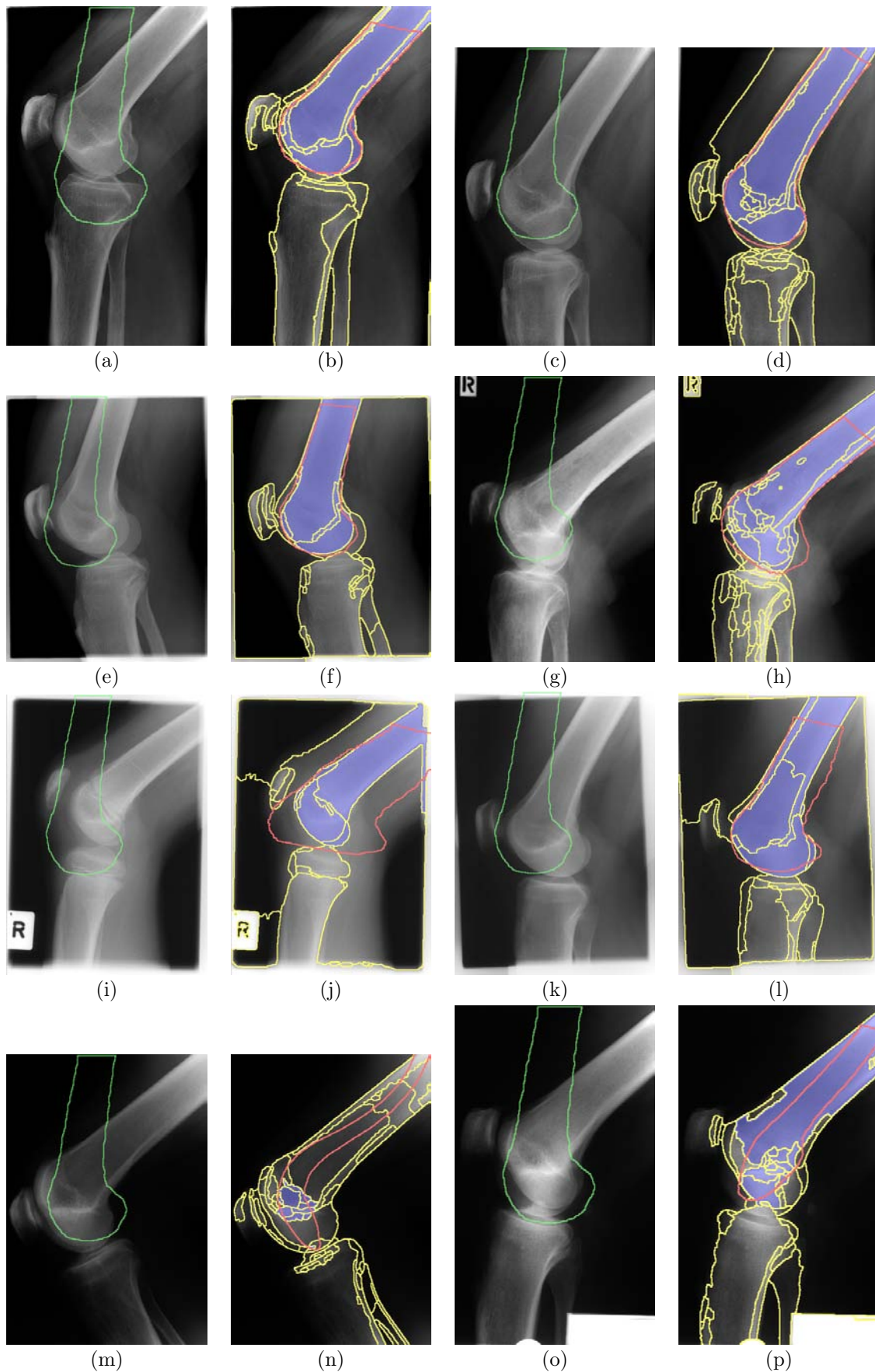




Rysunek 7.14: Wyniki detekcji drugiej kości śródreża lewej ręki z wykorzystaniem zaproponowanych w niniejszej pracy algorytmów łączenia segmentów (Rozdz. 6.2) i ścisłego dopasowania modelu do obiektu (Rozdz. 7.2) (opracowanie własne).

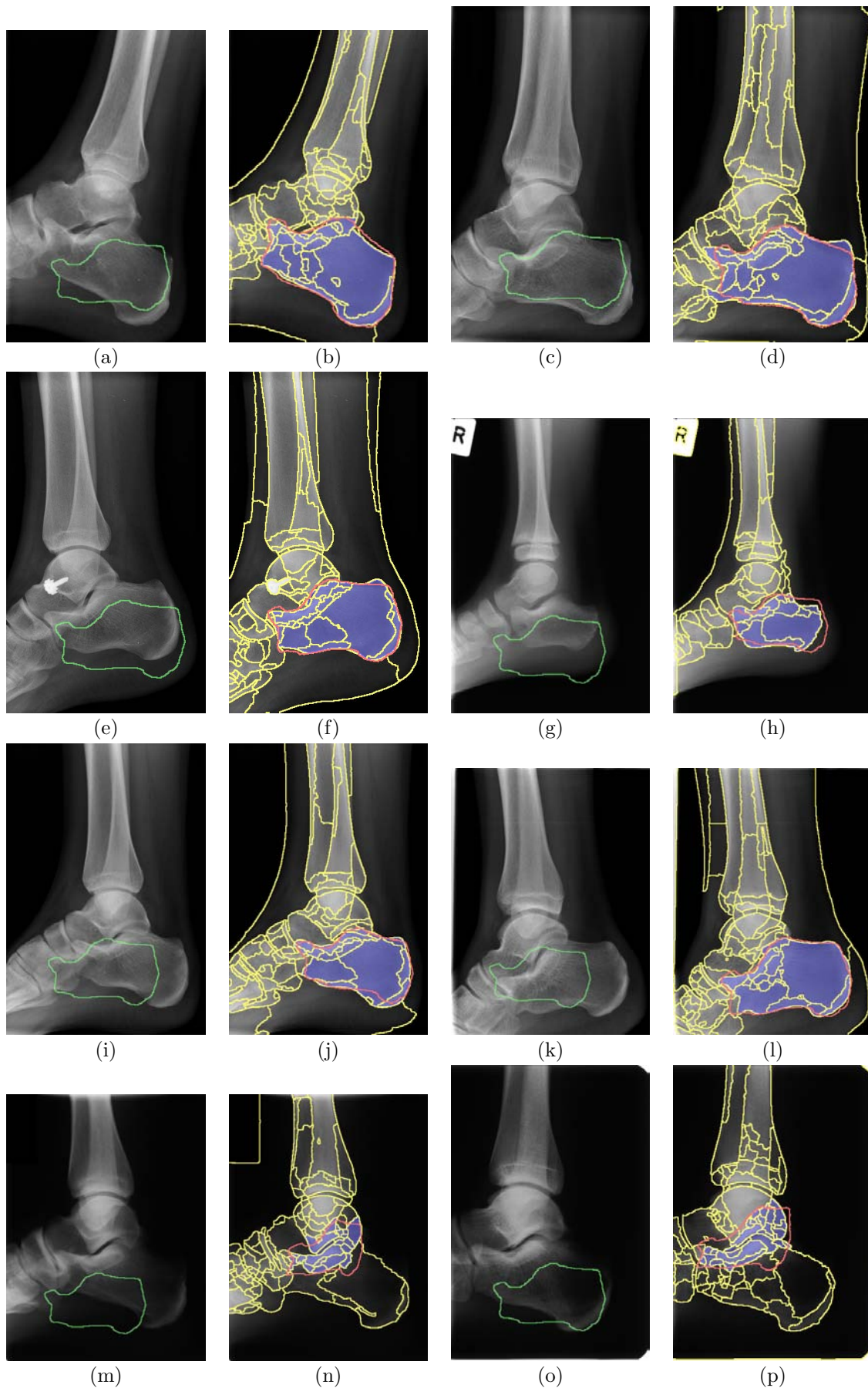


Rysunek 7.15: Wyniki detekcji kości piszczelowej z wykorzystaniem zaproponowanych w niniejszej pracy algorytmów łączenia segmentów (Rozdz. 6.2) i ścisłego dopasowania modelu do obiektu (Rozdz. 7.2) (opracowanie własne).

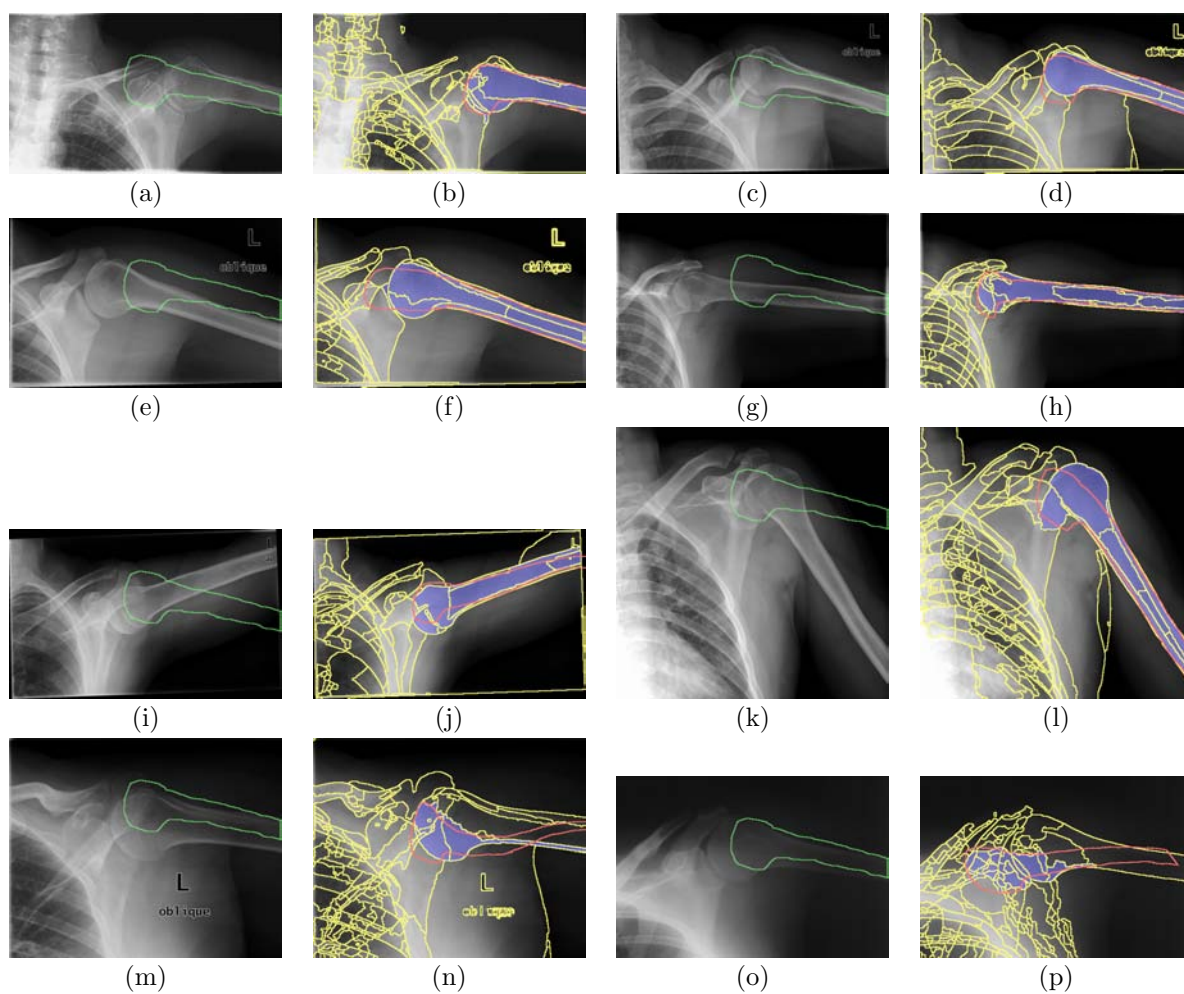


Rysunek 7.16: Wyniki detekcji kości udowej z wykorzystaniem zaproponowanych w niniejszej pracy algorytmów łączenia segmentów (Rozdz. 6.2) i ścisłego dopasowania modelu do obiektu (Rozdz. 7.2) (opracowanie własne).





Rysunek 7.17: Wyniki detekcji kości piętowej z wykorzystaniem zaproponowanych w niniejszej pracy algorytmów łączenia segmentów (Rozdz. 6.2) i ścisłego dopasowania modelu do obiektu (Rozdz. 7.2) (*opracowanie własne*).



Rysunek 7.18: Wyniki detekcji kości ramiennej z wykorzystaniem zaproponowanych w niniejszej pracy algorytmów łączenia segmentów (Rozdz. 6.2) i ścisłego dopasowania modelu do obiektu (Rozdz. 7.2) (*opracowanie własne*).

## **Podsumowanie wyników detekcji z użyciem modelu bazującego na kształcie**

W tym rozdziale opisano koncepcję detekcji dużych obiektów z wykorzystaniem szablonu obiektu przygotowanego przez użytkownika. Opiera się ona w znacznej mierze na znanej w literaturze metodzie segmentacji obrazu z wykorzystaniem atlasu. Dodatkowo został zaproponowany algorytm, którego celem jest poprawa wyników po etapie wstępnego nakładania obrazów. Opisana metoda może być z powodzeniem stosowana dla innych obrazów, niż wskazane w tym rozdziale obrazy medyczne. Jej użycie wymaga jednak każdorazowej oceny przez użytkownika – często popartej konsultacjami z ekspertem z danej dziedziny – czy spełnione są wstępne warunki jej użycia. Wymagają one, aby analizowany obiekt na poszczególnych zdjęciach wykazywał podobieństwo geometryczne dotyczące kształtu i rozmiaru oraz zachowywał stałość położenia względem innych obiektów. Nawet jeśli nie ma pewności czy aktualnie analizowane obrazy spełniają te warunki, warto przetestować opisaną metodę i dopiero kiedy wyniki okażą się niezadowolające przygotować dedykowany algorytm detekcji. Podsumowując, w sytuacji kiedy warunki użycia opisanej metody są spełnione, do głównych zalet jej użycia można zaliczyć:

- łatwość przygotowania nowych obrazów do analizy - użytkownik samodzielnie może przygotować proces detekcji poprzez zakreślenie obiektu na wybranym obrazie,
- ogólny sposób działania - metoda nie jest dedykowana dla jednego konkretnego przypadku i może być stosowana dla różnych obrazów.

### *Interaktywna metoda segmentacji obrazu bazująca na modelu znaczników kontrolnych*

Przedstawione w dwóch poprzednich rozdziałach algorytmy wspomagające segmentację oraz detekcję dużych obiektów działają w znacznej mierze w sposób automatyczny. Pierwszy z nich (Rozdz. 6) łączy małe obiekty powstałe w wyniku wstępnej segmentacji obrazu algorytmem działów wodnych w duże obiekty, opierając się na informacji o krawędziach zawartych między nimi. Użytkownik może wpływać na działanie algorytmu określając minimalną wartość kryterium łączenia. W drugim algorytmie (Rozdz. 7) użytkownik nie musi określać parametrów liczbowych<sup>1</sup>, a jedynie zakreśla obiekt do analizy i algorytm samodzielnie wykonuje detekcję na kolejnych obrazach. Ograniczeniem dla zastosowania tej metody<sup>2</sup> jest analiza obiektów o podobnym kształcie, które znajdują się w podobnym miejscu na obrazie. Kiedy jednak analizowane obiekty znacznie różnią się na poszczególnych obrazach, lub też analiza dotyczy różnych obiektów (np. detekcja zwierząt w ich naturalnych warunkach) może się okazać, że niezbędnym elementem będzie stała interakcja użytkownika z oprogramowaniem do analizy obrazu. Użytkownik mógłby wówczas indywidualnie wskazywać na każdym obrazie interesujące go obiekty, a komputer automatycznie wykonywałby niezbędne pomiary. Aby zmniejszyć wpływ czynnika ludzkiego (zmęczenie, brak staranności, brak doświadczenia, stres i inne), od którego może zależeć poprawność otrzymanych wyników, rola użytkownika w tym procesie powinna zostać ograniczona do minimum<sup>3</sup>. Algorytm wspomagający interaktywną segmentację obrazu powinien dawać użytkownikowi możliwość przekazywania informacji o interesujących go obiektach w jak najbardziej uogólnionej postaci, a także mimo nieznaczonej zmiany tej informacji, np. w przypadku powtórzenia analizy, powinien dawać takie same wyniki. Własność ta zwana *powtarzalnością procesu analizy obrazu*, znacznie zwiększa wiarygodność otrzymanych wyników.

W dalszej części tego rozdziału zostaną omówione możliwości użycia algorytmu działów wodnych do detekcji dużych obiektów w podejściu interaktywnym. Powszechnie znana

---

<sup>1</sup>Algorytm daje możliwość ustalenia parametru *sigma* rozmycia Gaussa, który wpływa na wielkość segmentów generowanych przez algorytm działów wodnych.

<sup>2</sup>Ograniczenia, o których mowa szerzej opisano na początku rozdziału 7.2.

<sup>3</sup>Tryb manualny, w którym użytkownik ręcznie zaznacza interesujące go obiekty powinien być stosowany wyłącznie wtedy, kiedy nie jest dostępny algorytm wspomagający detekcję lub jego przygotowanie nie jest uzasadnione z powodu bardzo małej liczby obrazów do analizy.

wersja tego algorytmu bazująca na tzw. znacznikach (ang. *Marker-based watershed image segmentation*) pozwala na przeprowadzenie szybkiej i dokładnej segmentacji obrazu, nie wymagając przy tym od użytkownika dokładnego zakreslenia poszukiwanych obiektów. Często wystarczy wskazanie kilku znaczników (punktów lub linii) kontrolnych, aby wyróżnić kluczowe obiekty z obrazu. Powtarzalność algorytmu jest bardzo stabilna i zmiana lokalizacji znaczników przy kolejnej analizie tego samego obrazu nie powoduje istotnej różnicy w podziale obrazu na segmenty<sup>4</sup>.

W rozdziale tym w odróżnieniu od dwóch wcześniejszych rozdziałów (6 i 7) nie są proponowane nowe algorytmy przetwarzania obrazu, a jedynie wskazywane są możliwości wykorzystania znanego algorytmu działów wodnych na potrzeby detekcji dużych obiektów. Informacją przekazywaną w tym wypadku do algorytmu o lokalizacji dużych obiektów będą znaczniki przygotowywane przez użytkownika w sposób interaktywny. Pokazane zostanie również jak za pomocą tego podejścia określać poziom szczegółowości detekcji (hierarchiczna segmentacja obrazu) i jak zadbać o poprawne połączenie dużych obiektów (znaczniki ciągłe).

## 8.1 Algorytm działów wodnych bazujący na znacznikach

Algorytm działów wodnych w swej podstawowej formie (klasyczny algorytm działów wodnych opisano w rozdziale 4.1) rozpoczyna działanie od lokalnych minimów i rozlewając się w ich obrębie wyznacza granice segmentów. Jego zmodyfikowana wersja pozwala na pominięcie części lokalnych minimów, a nawet określenie zupełnie dowolnych punktów startowych. W celu wyjaśnienia zmian wprowadzonych do algorytmu, ponownie można skorzystać z analogii do zalewania terenu przez wodę<sup>5</sup>. Jednak tym razem można założyć, że teren jest zupełnie nieprzepuszczalny dla wody. Aby zalewanie było możliwe w miejscach wskazanych przez nowe punkty startowe, w terenie są „wiercone” dziury aż do poziomu morza. Woda zaczyna wypływać (poziom wody się podnosi) na powierzchnię i w pierwszej kolejności zalewa lokalną dolinę. Następnie, jeśli sąsiednia dolina jest pusta, to woda zacznie się tam przelewać, a jeśli nie, to na styku dwóch zlewisk zostanie ustalona linia wododziału. Opisane podejście określane jest właśnie segmentacją wododziałową bazującą na znaczniku (ang. *Marker-based watershed segmentation*). Główną jego zaletą jest pewność, że liczba segmentów, która w rezultacie powstanie będzie taka sama jak liczba punktów startowych. Kluczowe w tym wypadku wyznaczenie punktów startowych

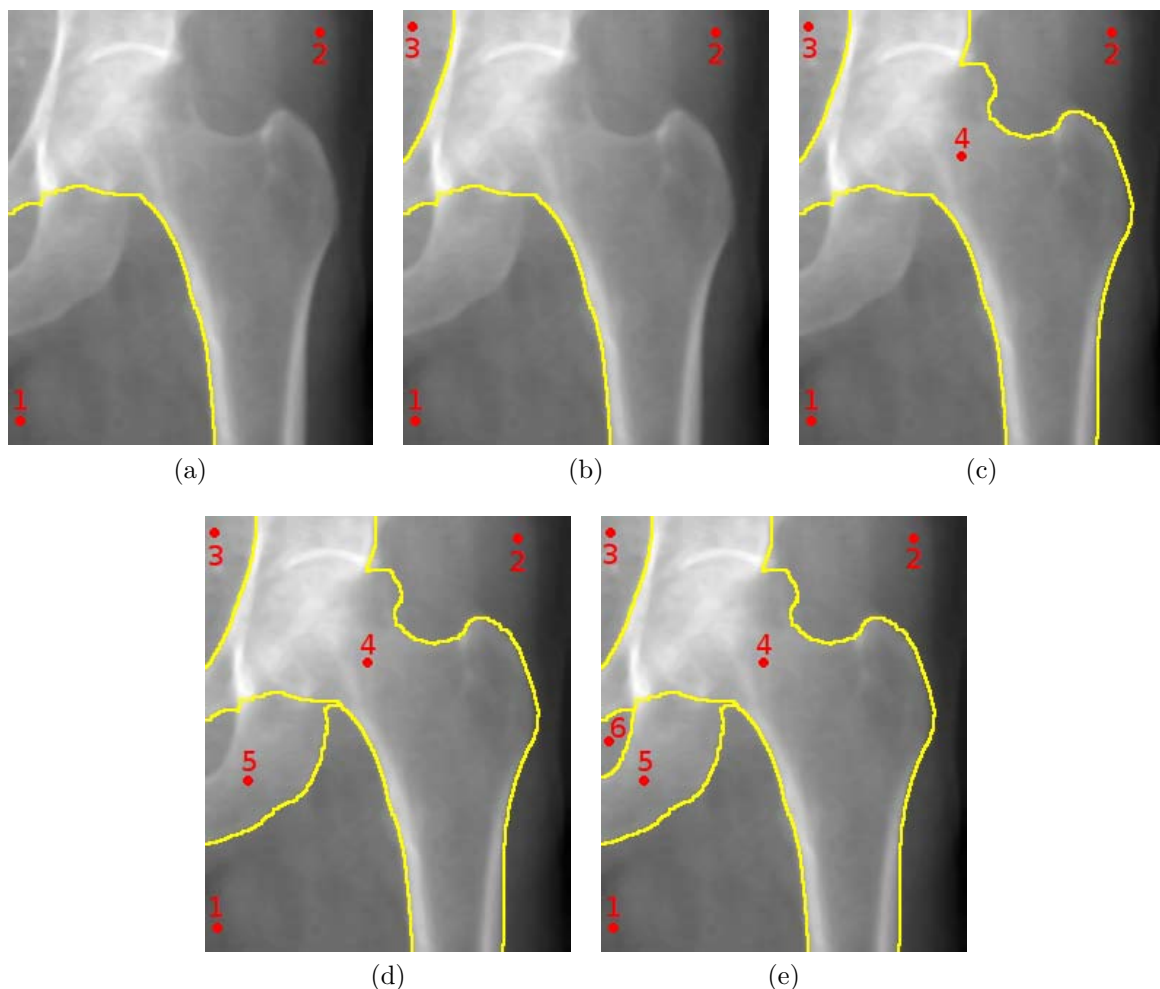
<sup>4</sup>Dobrą powtarzalność udało się uzyskać stosując znaczniki ciągłe. Stosowanie znaczników punktowych nie daje gwarancji powtarzalności segmentacji (przykład na rysunku 8.1 i 8.2; opis w tekście).

<sup>5</sup>Taka sama analogia została użyta w rozdziale 4.1, gdzie wyjaśniono podstawy działania algorytmu działów wodnych.



może wymagać przygotowania złożonych algorytmów detekcji, które nie dość, że muszą być dopasowywane do każdego przypadku osobno (np. inny algorytm do detekcji szyjki kości udowej, a inny do detekcji dwóch słoni na sawannie), to w dodatku nie daje pewności uzyskania pożądanego rezultatu (za dużo lub za mało punktów). W takiej sytuacji celowe może okazać się poproszenie użytkownika o ich wskazanie. Zostanie utracony w ten sposób czynnik automatyzacji obliczeń, ale będzie utrzymany ogólny charakter algorytmu i kontrola nad liczbą punktów startowych.

Na rysunku 8.1 pokazano działanie algorytmu działów wodnych ze znacznikiem. Użytkownik za pomocą interaktywnego programu zaznacza kolejne punkty zarówno dla tła jak i obiektu, a program sam wyznacza odpowiednie segmenty. Analizując przedstawiony przykład można odnieść wrażenie, że algorytm działa zgodnie z intencją użytkownika i pozwala na kontrolowaną przez niego segmentację poszczególnych obiektów.



Rysunek 8.1: Interaktywna segmentacja obrazu algorytmem działów wodnych z zastosowaniem znaczników punktowych (*opracowanie własne*).

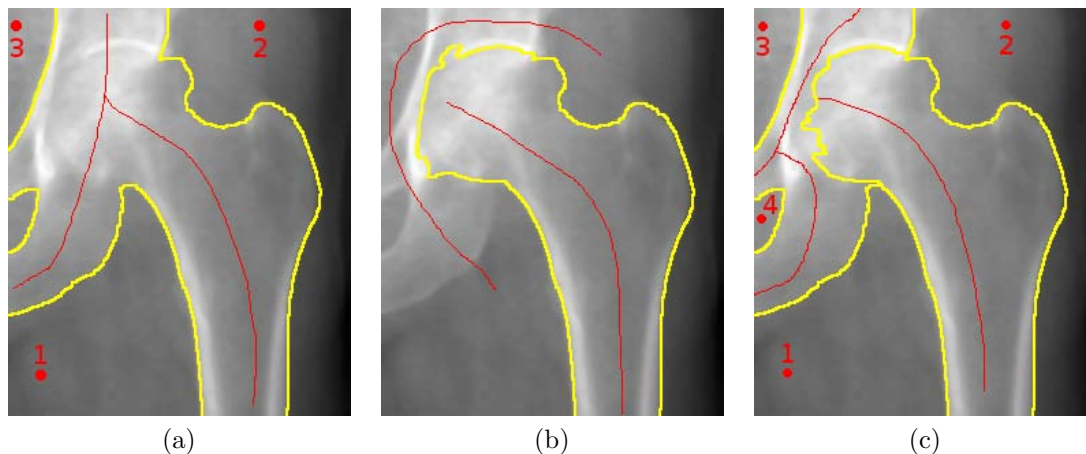
Niestety, opisana koncepcja pomimo swoich zalet nie zawsze daje pożądane efekty. Z przeprowadzonych testów wynika, że położenie punktu startowego ma wpływ na wynik segmentacji. Na rysunku 8.2 pokazano wyniki segmentacji po zmianie lokalizacji punktu numer 4. Rezultaty w tym wypadku są dalekie od oczekiwanych i znacznie



Rysunek 8.2: Wpływ położenia punktu startowego na wynik segmentacji obrazu algorytmem działów wodnych (*opracowanie własne*).

różnią się od wyników uzyskanych poprzednio (Rys. 8.1c). Powodów tego stanu rzeczy może być wiele. Najczęściej jest to związane z wysokim poziomem szumu, który trudno usunąć filtrami wygładzającymi. Na wynik wpływa także duża niejednorodność obiektów i występowanie mocnych wewnętrznych granic przecinających obiekt. Inną przyczyną, nie związaną z jakością obrazu, jest problem z interpretacją informacji na nim zawartej. Nawet najlepszy algorytm segmentacji obrazu nie jest w stanie bezbłędnie rozdzielić obiektów na podstawie bardzo szczątkowych informacji (wskazanie jednego punktu może okazać się niewystarczające). Trudno również oczekiwać, aby algorytm odgadł intencje użytkownika w przypadku obrazów, gdzie detekcja nie jest jednoznaczna (np. w przypadku przylegających kilku identycznych obiektów).

Aby wyeliminować opisane przed chwilą problemy, zalecane jest bardziej szczegółowe określanie obiektów do analizy. Można to osiągnąć prosząc użytkownika o zaznaczenie nie tylko punktów startowych, ale również linii zawartych w obiekcie lub przecinających kilka obiektów. W ten sposób algorytm zostanie rozszerzony o tzw. *znaczniki ciągłe*, które znacznie zwiększają szanse uzyskania poprawnej segmentacji. Odnosząc się ponownie do przykładu związanego z zalewaniem terenu, tym razem nie będzie wiercona dziura do poziomu morza, a zostanie wykopany długi rów. Woda wylewając się z niego będzie zalewać przyległe doliny i łączyć je natychmiast ze sobą kanałem przecinającym wzgórze. Podejście to zilustrowano na kilku przykładach na rysunku 8.3. Widać tym razem, że użytkownik ma lepszą kontrolę nad przebiegiem procesu segmentacji i może dokładnie określić, co jest celem analizy obrazu (np. wydzielenie obszaru wszystkich



Rysunek 8.3: Możliwości dokładnego określania obiektów do segmentacji z zastosowaniem znaczników ciągłych (*opracowanie własne*).

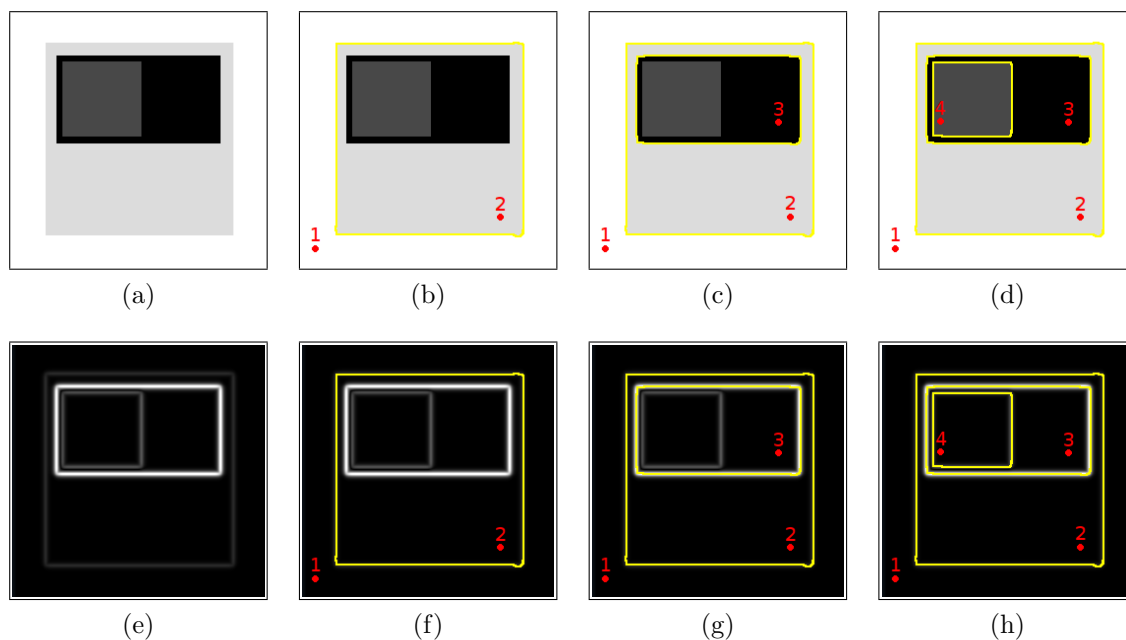
kości od tła (Rys. 8.3a), wydzielenie obszaru kości udowej od pozostałych elementów (Rys. 8.3b), wydzielenie obszaru kości udowej, miednicy i tła (Rys. 8.3c)). Główną zaletą tego podejścia jest brak konieczności dokładnego definiowania znaczników – wystarczą przybliżone linie, aby uzyskać pożądaną rezultat. Własność ta, w odróżnieniu od innych dodatkowo przetestowanych podejść interaktywnych - metody aktywnej linii (ang. *Live-wire technique*)<sup>6</sup> i prostym algorytmom rozrostu obszaru (ang. *Region growing*)<sup>7</sup> – pozwala na uzyskanie podobnych, a nawet identycznych wyników przez osoby, które niezależnie zaznaczają znaczniki startowe. W związku z tym, że opisywana metoda jest metodą interaktywną, w przypadku pojawienia się błędów w segmentacji można dokładniej doprecyzować położenie znaczników.

## 8.2 Hierarchiczna segmentacja obrazu

Po dokładnym przeanalizowaniu przykładu z rysunku 8.1 można zauważyć, że wskazywanie kolejnych punktów startowych nie zmienia wcześniej wyznaczonych linii wododziału, a proces segmentacji dotyczy wyłącznie obszaru, na którym ponownie zaznaczono punkt startowy. Takie działanie algorytmu pozwala na podział większych obiektów na mniejsze, bez zmiany już ustalonych granic. Podejście to umożliwia hierarchiczne rozdzielanie obiektów na coraz mniejsze części składowe w zależności od wymaganego stopnia szczegółowości [55].

<sup>6</sup>Metoda aktywnej linii (ang. *Live-wire technique*), która wymaga precyzyjnego wskazania zbioru punktów położonych blisko siebie na obwodzie poszukiwanego obiektu.

<sup>7</sup>Działanie algorytmów rozrostu obszaru (ang. *Region growing*) jest nie tylko uzależnione od punktów startowych, ale również od wartości granicznej decydującej o asymilacji kolejnych punktów.

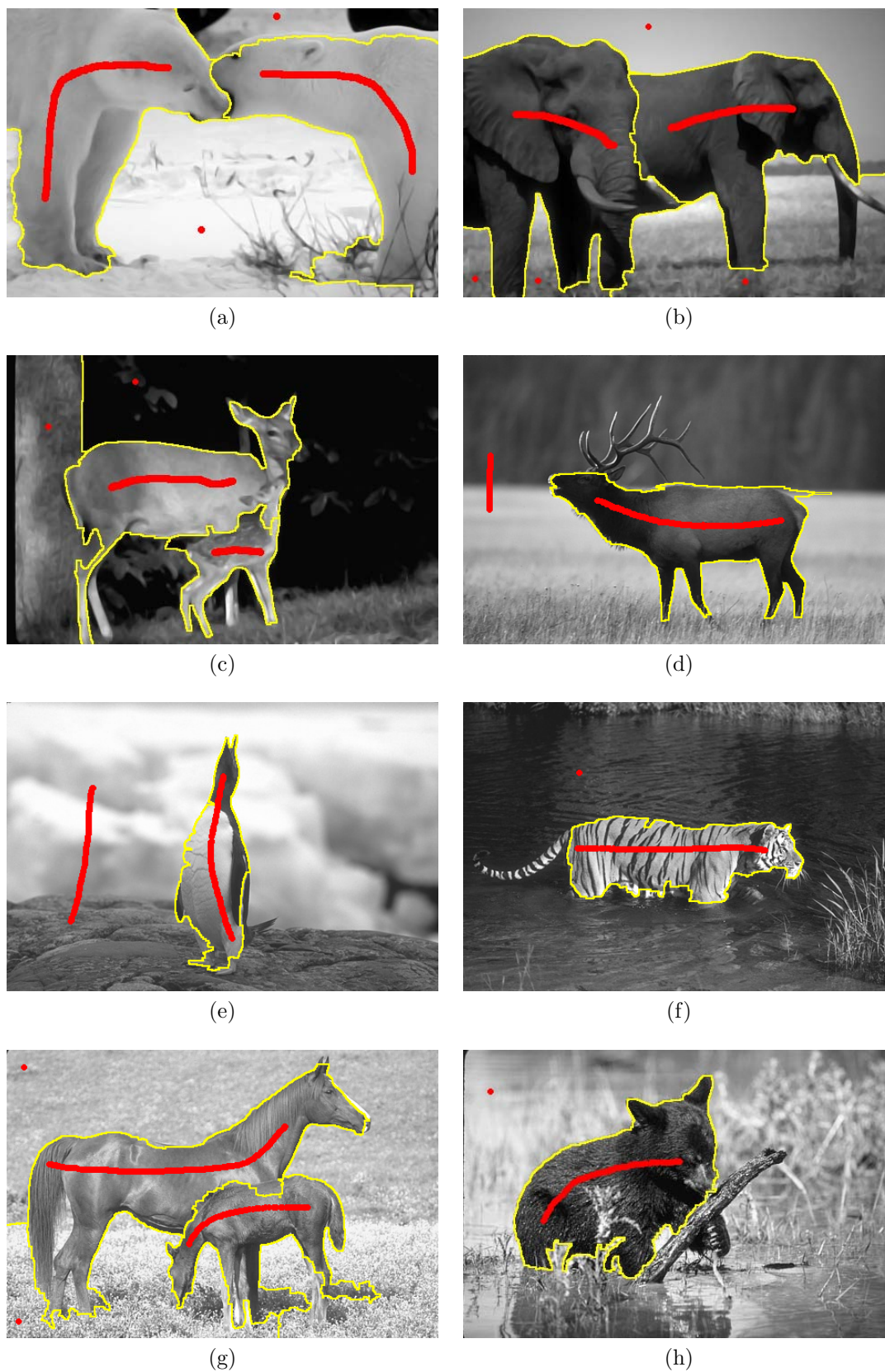


Rysunek 8.4: Proces hierarchicznej segmentacji obrazu z wykorzystaniem algorytmu działów wodnych na sztucznym obrazie (w dolnym wierszu obraz gradientowy) (opracowanie własne).

W celu dokładniejszego wyjaśnienia w jaki sposób korzystać z omawianego algorytmu, aby osiągnąć pożądane efekty, zostanie dokładnie przeanalizowany sposób jego działania na sztucznym obrazie. Na rysunku 8.4a widać kwadrat, na którym zostały narysowane mniejsze elementy (prostokąt, a w nim kolejny kwadrat). Poniżej znajduje się odpowiadający mu obraz gradientowy (Rys. 8.4e). Dwa pierwsze punkty startowe zostały zaznaczone w celu oddzielenia obiektu od tła (Rys. 8.4b). Warto zwrócić uwagę, iż pomimo wyraźnej granicy między czarnym prostokątem a kwadratem, w którym ten prostokąt znajduje się, został on poprawnie zaliczony w jego obszar. Stało się tak, ponieważ po zdefiniowaniu linii wododziału między obszarem pierwszym i drugim, czarny prostokąt mógł zostać zalany tylko z obszaru, który bezpośrednio z nim graniczy, a jest nim obszar drugi, czyli kwadrat. Podobna sytuacja ma miejsce dla kolejnych dwóch punktów startowych (Rys. 8.4c i 8.4d). Tutaj dodatkowo widać, że dodanie ich nie zmienia wcześniej ustalonych granic, a jedynie rozdziela segmenty, na których był już zaznaczony punkt startowy. Odnosząc się ponownie do analogii związanej z zalewaniem terenu, łatwo można zauważyć, że wywiercenie kolejnej dziury w już zalanym obszarze nie może zmienić jego obszaru zalewnia, a jedynie wprowadzić wewnątrz niego kolejny podział.

## 8.3 Podsumowanie

Opisany algorytm działów wodnych bazujący na znacznikach pozwala na skonstruowanie oprogramowania, które może znacznie ułatwić i przyspieszyć proces poprawnej segmentacji obrazu. Dzięki interaktywnemu wskazywaniu znaczników startowych przez użytkownika zachowana jest kontrola nad poprawnością wyników, a sam użytkownik może dowolnie określać stopień szczegółowości detekcji. W odróżnieniu od znaczników punktowych, zastosowanie znaczników ciągłych pozwoliło na dokładne określenie obiektów, które mają zostać wydzielone z obrazu. Na rysunku 8.5 można zobaczyć kolejne przykłady skuteczności przedstawionej koncepcji.



Rysunek 8.5: Przykłady działania algorytmu działów wodnych ze znacznikiem na obrazach przedstawiających naturę (*opracowanie własne*).

# Część IV

## Podsumowanie



### *Propozycje rozszerzające możliwości proponowanej metody*

W III części pracy omówiono koncepcje wyznaczania położenia dużych obiektów na obrazach niejednorodnych. Bazując na znanych metodach nakładania obrazu i autorskim algorytmie dokładnego dopasowania modelu do obiektu (Rozdz. 7) możliwa jest detekcja dużych obiektów o podobnym kształcie. Metoda ta w podstawowej postaci korzysta z wyników segmentacji uzyskanych algorytmem działów wodnych. Jeśli jednak otrzymane rezultaty odbiegają od oczekiwanych, można zastosować dodatkowy algorytm łączenia segmentów, które przedstawiono w rozdziale 6. Połączenie tych dwóch algorytmów pozwoliło na poprawną detekcję kości udowej i innych analizowanych obrazów (Rozdz. 7.2), podczas gdy powszechnie znane algorytmy segmentacji obrazu nie dawały tak dobrych wyników. Dodatkowo w rozdziale 8 pokazano możliwość użycia algorytmu działów wodnych ze znacznikiem w podejściu interaktywnym do poprawy ewentualnych błędnych detekcji. Podejście to również polecono do detekcji obiektów, które znacznie różnią się kształtem i położeniem na obrazie. Nie wymaga ono precyzyjnego zakreślania obiektów badań, a jedynie przybliżonego wskazania ich obszaru zalegania.

Wadą proponowanej w rozdziale 7 koncepcji detekcji dużych obiektów, która nie była szczegółowo omówiona, jest jej długi czas wykonania. Zależy on przede wszystkim od wybranego algorytmu nakładania obrazu. W pracy zdecydowano się na użycie optymalizacyjnego algorytmu zezwalającego na deformację obrazu w trakcie nakładania [3]. Dla przykładowych analizowanych obrazów czas wyliczenia wektora transformacji jednego obrazu w drugi i późniejszej deformacji wzorcowego modelu wynosił odpowiednio<sup>1</sup>: dla jednego obrazu klatki piersiowej<sup>2</sup> ok. 11s, a dla kości udowej<sup>3</sup> ok. 7s. Nawet jeśli dla serii składającej się z kilku zdjęć czas analizy będzie akceptowalny, to już dla 100 obrazów może być on uznany za zbyt długi, ponieważ czas przetwarzania wynosi odpowiednio 18 i 12 minut. Częściowym rozwiązaniem tego problemu może być zapisywanie wyników po raz przeprowadzonej analizie obrazu. Zapis tym bardziej

---

<sup>1</sup>Testy obciążeniowe prowadzone były na komputerze klasy PC z procesorem AMD taktowanym z częstotliwością 2,8GHz. W trakcie obliczeń używany był tylko jeden z czterech dostępnych rdzeni obliczeniowych procesora. Dostępny rozmiar pamięci operacyjnej wynosił 4GB. System operacyjny: *Linux Ubuntu* w wersji 64 bitowej.

<sup>2</sup>Obrazy płuc miały wielkość  $512 \times 512$  pikseli kodowanych liczbą o wartości od 0 do 255. Daje to wielkość pliku równą 256KB.

<sup>3</sup>Obrazy kości udowej miały wielkość  $251 \times 299$  pikseli kodowanych liczbą o wartości od 0 do 255. Daje to wielkość pliku równą 73,29KB.

jest słuszny, jeżeli dla powstałych błędów wprowadzane były jakiegokolwiek korekty przez użytkownika, lub była przeprowadzana interaktywna segmentacja obrazu. W ten sposób praca raz wykonana nie będzie musiała być ponownie powtarzana i będzie mogła być dodatkowo zweryfikowana bądź poprawiona przez innego użytkownika. Poza obrazami wynikowymi zarchiwizowane powinny być również wszystkie dane liczbowe i uwagi osoby przeprowadzającej daną analizę. Gromadzenie tego typu informacji może okazać się szczególnie przydatne do przeprowadzania analiz porównawczych, a w przypadku diagnostyki medycznej usprawni śledzenie historii choroby.

## 9.1 Projekt automatycznego systemu rozpoznawania i detekcji dużych obiektów

W celu zminimalizowania niedogodności, jaką jest długi czas obliczeń, po przeprowadzonej analizie obrazu powinny być utrwalone zarówno uzyskane wyniki liczbowe, jak i wnioski osoby przeprowadzającej analizę, a także końcowe i jeśli będzie taka potrzeba - pośrednie obrazy wygenerowane przez użyte algorytmy. W tym celu autor niniejszej rozprawy rozpoczął prace nad projektem **graficznej bazy danych**, który będzie zawierał przedstawione w rozdziale 6, 7 i 8 algorytmy i metody detekcji dużych obiektów. Pomimo planowanego ogólnego charakteru opracowywanego systemu, będzie on mógł znaleźć szczególne zastosowania do wsparcia diagnostyki medycznej. W chwili obecnej system w wstępnej fazie projektowej znajduje się na etapie implementacji<sup>4</sup>. Aby umożliwić jego jak najszerszą popularyzację i zarazem zachęcić inne osoby zainteresowane podobną tematyką do prac badawczo-rozwojowych, program udostępniono na zasadach otwartej *licencji BSD*<sup>5</sup>. W dalszej części rozdziału zostanie opisana koncepcja systemu oraz możliwości jego użycia na potrzeby diagnostyki medycznej.

## 9.2 Koncepcja systemu

W chwili obecnej<sup>6</sup> głównym celem systemu jest wsparcie użytkownika w obsłudze zaproponowanych w części III metod detekcji dużych obiektów. Program będzie koncentrował się przede wszystkim na jak największej automatyzacji wykonywanych czynności oraz na archiwizacji wyników. Zadaniem, które sobie postawiono, jest takie działanie systemu, aby użytkownik nie musiał wykonywać powtarzających się operacji, np.

<sup>4</sup>Aktualny stan prac nad projektem można zobaczyć na stronie WWW [87] autora niniejszej rozprawy.

<sup>5</sup>Licencja zezwala na dostęp do kodu źródłowego z możliwością modyfikacji i dowolnego użytku.

<sup>6</sup>Stan programu na 2011 rok. Aktualny stan systemu można sprawdzić na stronie WWW [87].

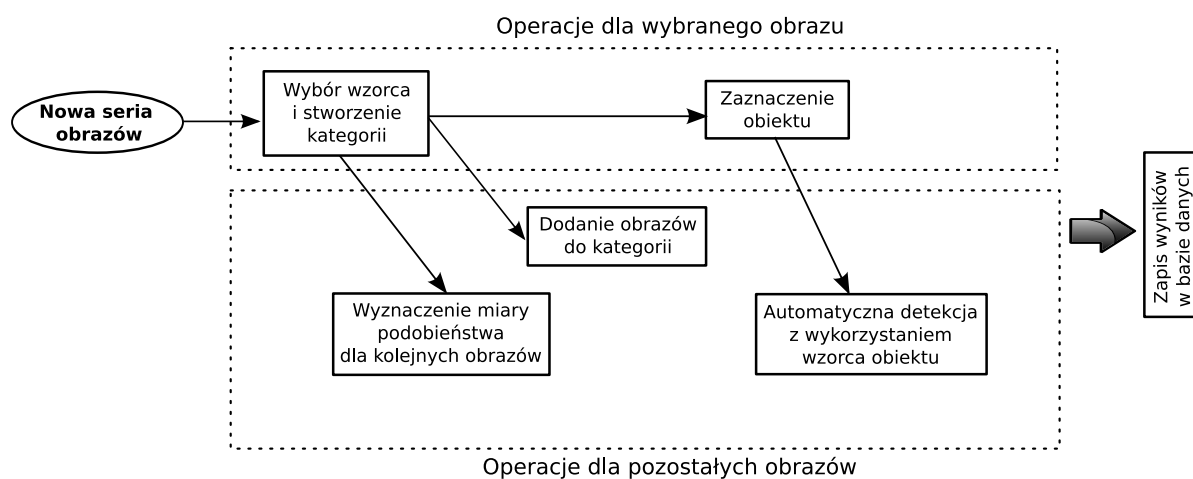
wskazywać punkty startowe i często określać parametry wejściowe, ale mógł nadzorować poprawność działania programu.

Wiele z obecnie dostępnych graficznych baz danych koncentruje się na analizie obrazów bez wnikania w ich treść znaczeniową. Analizowane są wówczas albo cechy globalne takie jak kolor i tekstura, albo też lokalne dotyczące wybranych regionów. Regiony te mogą być manualnie zaznaczane przez użytkownika, bądź też wyznaczane z pomocą dedykowanych algorytmów. W większości są to systemy obsługujące tylko jeden typ obrazów, a mechanizmy analityczne w nich zawarte dotyczą specyficznego problemu. Konstruowanie osobnych systemów do analizy różnych typów zdjęć i dla każdego problemu analitycznego może być nie tylko kosztowne i wymagać sporo czasu, ale może też wprowadzać chaos w ich obsłudze (różne interfejsy użytkownika) i problem z integracją danych w nich zawartych. Ideałem byłoby konstruowanie takich systemów, które korzystając z treści znaczeniowej zawartej w obrazie wspierałyby analizę różnego rodzaju obrazów. Wydaje się, że w chwili obecnej szansą na realizację tego typu systemów są metody automatycznego rozumienia obrazu (ang. *Automatic Image Understanding*) [70, 49], w których dzięki semantycznemu opisowi obrazów możliwa jest analiza zależności pomiędzy poszczególnymi obiektami obrazu. Tym samym łatwiejsze staje się poszukiwanie przypadków, dla których te relacje zostały zakłócone, np. patologie na zdjęciach medycznych. Planowane w projektowanym systemie funkcje nie wybiegają aż tak daleko, jednak dzięki powiązaniu informacji zawartej w obrazie i dostępnych metadanych tekstowych będzie możliwa szybka analiza wybranych obiektów. Dodatkowo zakłada się, że system będzie obsługiwał różne typy obrazów, chociaż w obecnej chwili użycie algorytmu detekcji opisanej w rozdziale 7 wymusza, aby obiekty w każdej analizowanej serii zdjęć miały podobny kształt i znajdowały się w zbliżonym miejscu na obrazie. Dzięki otwartej architekturze systemu i dostępowi do kodu źródłowego możliwe jest jednak dodawanie kolejnych typów algorytmów detekcji, przez co zakres stosowania systemu może się zwiększać.

Zakłada się, że projektowana graficzna baza danych po wstępnym etapie konfiguracji będzie mogła działać w sposób automatyczny. Aby było to możliwe, obrazy powinny zostać opisane i powinien zostać wskazany cel analizy. Niezbędne informacje mogą zostać podane przez użytkownika lub też odczytane z metadanych zawartych w pliku graficznym, np. informacje o pacjencie i badaniu z obrazów *DICOM*<sup>7</sup>. Następnie zaleca się stworzenie

<sup>7</sup>DICOM, ang. *Digital Imaging and Communications in Medicine* (Obrazowanie Cyfrowe i Wymiana Obrazów w Medycynie) - norma opracowana przez ACR/NEMA (*American College of Radiology / National Electrical Manufacturers Association*) dla potrzeb ujednoczenia wymiany i interpretacji danych medycznych związanych lub reprezentujących obrazy diagnostyczne w medycynie [91].

kategorii dla nowo dodawanej serii obrazów i wybranie z niej obrazu wzorcowego<sup>8</sup>. Na tym obrazie manualnie lub też z wykorzystaniem metod interaktywnych (np. metody opisanej w rozdziale 8) należy kolejno zaznaczać obiekty badań (np. lewe i prawe płuco na obrazie tomograficznym klatki piersiowej) i nadać im tekstowe etykiety (np. „*lewe płuco*”, „*prawe płuco*”). Dzięki temu możliwe będzie przygotowanie bardziej naturalnych dla człowieka mechanizmów wyszukiwania. Np. będzie można zrealizować zapytanie typu: „*pokaż obrazy klatki piersiowej, na których lewe płuco jest większe od płuca prawego*”, lub inne w zależności od rodzaju dostępnych pomiarów. Korzystając z metody opisanej w rozdziale 7, obrys dla jednego obrazu posłuży do detekcji obiektów na pozostałych obrazach należących do tej samej kategorii. Poglądowy diagram dla tego etapu pracy systemu przedstawiono na rysunku 9.1. Zawiera on uproszczone bloki przedstawione w rozdziale 7 na rysunkach 7.2, 7.3 i 7.4.



Rysunek 9.1: Projekt graficznej bazy danych - pierwsze dodanie obrazów do bazy (*opracowanie własne*).

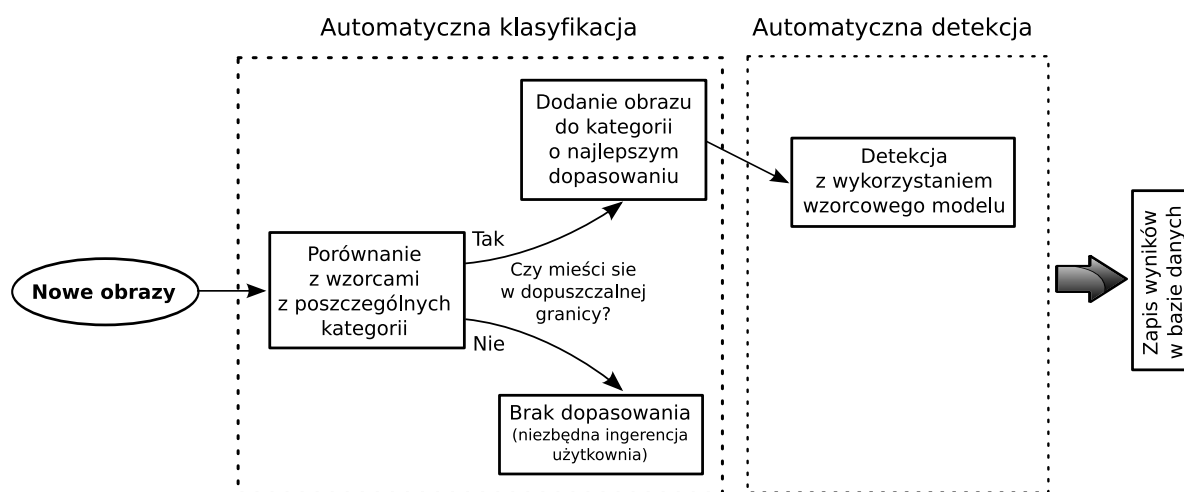
Jak wspomniano wcześniej, podczas indeksowania danych w bazie poza informacją tekstową będzie również uwzględniana informacja graficzna. Dla każdego obrazu będzie wyznaczana miara podobieństwa do obrazu wzorcowego i wartość tej miary dodatkowo będzie zapisywana w bazie danych. Wśród konstruowanych obecnie graficznych baz danych, zazwyczaj używa się informacji o kolorze [14] [35], teksturze [43] [74], kształcie [29] [45] i położeniu obiektu [54] [17]<sup>9</sup>. Na chwilę obecną w prototypie systemu zaimplementowano algorytm opisany w pracy [31], korzystający z metod analizy falkowej (użyto falki *Haara*) do analizy tekstury obrazu<sup>10</sup>.

<sup>8</sup>Obraz wzorcowy rozumiany jest w tym wypadku jako obraz, na którym poszukiwany obiekt ma typowe kształty. Obraz ten może zostać wybrany subiektywnie przez użytkownika. Rozważa się możliwość wyboru kilku obrazów, które będą reprezentować daną grupę.

<sup>9</sup>Aktualny stan prac w zakresie metod porównywania, indeksacji i wyszukiwania obrazów można znaleźć w [64, 36, 46].

<sup>10</sup>Implementacja przy znacznym wykorzystaniu kodu z programu *imgSeek* [83].

Aby lepiej wyjaśnić koncepcję działania systemu zostanie przedstawiony **przykładowy scenariusz jego użycia**. Załóżmy, że istnieje potrzeba dodania do bazy danych zdjęć tomograficznych przedstawiających klatkę piersiową na potrzeby analizy zmian zachodzących w płucach. W pierwszym momencie liczba tych zdjęć nie jest duża i wynosi najwyżej kilkanaście. Użytkownik wykonuje czynności przedstawione na diagramie 9.1 - tworzy kategorię, wybiera dla niej etykietę (np. „*zdjęcia tomograficzne klatki piersiowej*”, lub skrótowo „*płuca*”), którą zostaną oznaczone wszystkie zdjęcia. Następnie na wybranym obrazie zaznacza prawe i lewe płuco, a system automatycznie wykonuje detekcję na kolejnych obrazach. Każdy obraz jest indeksowany według wybranej miary podobieństwa (lub kilku miar, jeśli są dostępne) i wartość indeksu (dopasowania) wraz z wynikiem detekcji zapisywana jest w graficznej bazie danych. Podobną czynność można przeprowadzić dla kolejnych rodzajów zdjęć, np. zdjęć radiologicznych kości udowej. Po wykonaniu tych czynności można uznać, że **system skonfigurowano i będzie działał w sposób automatyczny**, dla kolejnych zdjęć należących do kategorii, które już istnieją w bazie. Czynności, które będą wówczas wykonywane przedstawiono na diagramie 9.2. Dodawanie obrazów w tym wypadku może odbywać



Rysunek 9.2: Projekt graficznej bazy danych - dodanie kolejnych obrazów do bazy (*opracowanie własne*).

się już bez ingerencji użytkownika. System może automatycznie monitorować pewien zasób, np. dysk sieciowy lub katalog na dysku lokalnym i dodawać wszystkie obrazy, które zostały tam umieszczone. Następnie każdy obraz porównywany jest ze wzorcowymi obrazami, zapisanymi wcześniej w bazie i dodawany jest do tej kategorii, dla której uzyskał najlepsze dopasowanie do reprezentującego ją wzorca. Zaleca się dodatkowo określenie minimalnej wartości progowej, która jeśli nie zostanie osiągnięta podczas wyliczania miary podobieństwa, to wówczas taki obraz nie zostaje sklasyfikowany i powinien oczekiwać na decyzję użytkownika. Po przypisaniu obrazu do kategorii następuje automatyczna

detekcja obiektów z wykorzystaniem modeli obiektów zapisanych w bazie - np. dla obrazów tomograficznych klatki piersiowej będzie to obszar płuc.

## 9.3 Realizacja systemu

Implementacja zaproponowanej koncepcji graficznej bazy danych jest na etapie realizacji. Do tej pory powstał już działający prototyp, który oferuje część z zaplanowanych funkcji. Postęp prac można śledzić na stronie autora [87], gdzie również dostępne są kody źródłowe do pobrania. Jako język programowania wybrano C++ ze względu na jego wysoką wydajność. Biblioteką, która wspiera operacje przetwarzania obrazu jest *CImg* [89]. Biblioteka ta również obsługuje wyświetlanie interfejsu użytkownika oraz interakcje za pomocą klawiatury i myszki. Po etapie zakończenia prac nad prototypem rozważane jest skorzystanie z bardziej zaawansowanej biblioteki *GUI* (ang. *Graphical User Interface*) takiej jak *Qt* [86], lub *GTK+* [81]. System składa informacje zarówno tekstowe jak i graficzne w lokalnej bazie danych opartej na *SQLite* [88]. Po zakończonym etapie przygotowania prototypu planowane jest zastosowanie modelu klient-serwer, co pozwoli na jednoczesną pracę wielu użytkowników na tej samej bazie.

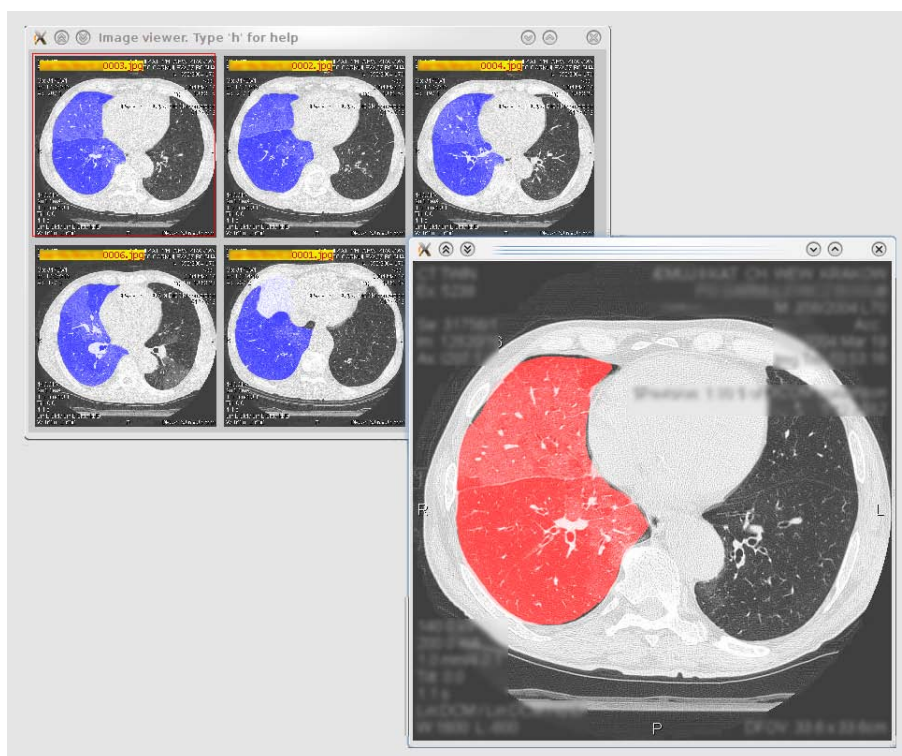
Poniżej wyliczono listę funkcji oferowanych przez system:

- Możliwość seryjnego wczytywania obrazów po podaniu listy ścieżek do plików,
- Odczyt formatu *DICOM* służącego do składowania zdjęć medycznych,
- Składowanie obrazów i wyników detekcji w plikowej bazie danych,
- Przeglądarka grafiki wyświetlająca obrazy z bazy danych oraz wyniki detekcji (Rys. 9.3),
- Indeksacja obrazów z użyciem falek Haara,
- Wyszukiwanie zbioru obrazów najbardziej podobnych do wskazanego obrazu wzorcowego. Obraz może zostać wskazany interaktywnie z obrazów widocznych w przeglądarce (kliknięcie środkowym przyciskiem myszy) lub może zostać określony jako parametr podczas startu systemu.
- Klasyfikacja obrazów na grupy tak, jak przedstawiono to na diagramie 9.2,
- Opracowany w niniejszej pracy algorytm detekcji dużych obiektów z wykorzystaniem wzorca kształtu<sup>11</sup>.

---

<sup>11</sup>W dostępnej na stronie internetowej [87] autora niniejszej pracy implementacji systemu wskazany algorytm zostanie dodany w chwili upublicznienia prowadzonych wyników prac w postaci artykułu naukowego, bądź też publicznej obrony rozprawy doktorskiej.





Rysunek 9.3: Przykładowy zrzut ekranu przedstawiający działanie skonstruowanej graficznej bazy danych (*praca własna*).

Funkcje systemu, nad którymi obecnie trwają prace i które są planowane do zrealizowania w najbliższym czasie:

- Odczyt metadanych z pliku *DICOM*,
- Inne metody indeksowania obrazów, np. bazujące na kolorze lub teksturze,
- Rozszerzenie języka SQL o możliwość wyszukiwania na podstawie informacji graficznej,
- Próby wdrożenia metod syntaktycznych odnoszących się do koncepcji automatycznego rozumienia obrazów [70, 49],
- Zmiana architektury z lokalnej na architekturę typu klient-serwer;
- Możliwość podłączenia się do zewnętrznych obrazowych baz danych, np. medycznych systemów *PACS*<sup>12</sup>. Pozwoli to na skorzystanie z dostępnych w projektowanym systemie algorytmów detekcji dużych obiektów, bez zbędnego powielania danych.

---

<sup>12</sup>PACS, ang. *Picture Archiving and Communication Systems* - systemy archiwizacji i przesyłania obrazów medycznych.



## 9.4 Podsumowanie prac nad projektem

Połączenie możliwości graficznej bazy danych i opracowanych w III części pracy algorytmów segmentacji i detekcji dużych obiektów pozwoliło na przygotowanie prototypu systemu, który już teraz zdaje się potwierdzać elastyczność i wygodę ich użycia. Dzięki temu, że system obsługuje różnego typu obrazy i nie jest dedykowany do rozwiązywania wyłącznie jednej klasy problemów, możliwa jest szybka jego konfiguracja w celu przeprowadzenia nowego typu analiz. System obsługuje detekcję obiektów o zbliżonym kształcie, znajdujących się w podobnym miejscu na obrazie. Dlatego wystarczy, że użytkownik zaznaczy obiekt na jednym obrazie, a system sam wykona jego detekcję na kolejnych. Ta cecha jest niezwykle istotna dla użytkownika, ponieważ pozwala mu skoncentrować się na problemie i nie wymaga od niego znajomości procedur przetwarzania obrazu. Tryb obsługi systemu, który zaproponowano, kładzie duży nacisk na automatyzację wykonywanych czynności, dzięki temu po etapie wstępnej konfiguracji praktycznie nie wymaga ingerencji użytkownika w jego działanie. Oczywiście nie można zakładać jego bezbłędnego funkcjonowania, dlatego zaleca się okresową kontrolę wykonanych przez system detekcji obiektów, np. po dodaniu większej liczby zdjęć. Obecnie prowadzone i planowane przez autora prace stawiają sobie za cel udoskonalenie mechanizmów wyszukiwania i klasyfikacji obiektów, jak również wdrożenie różnych metod detekcji obiektów.

### *Podsumowanie i wnioski*

W prezentowanej pracy podjęto próbę przedstawienia możliwości, jakie dają metody komputerowej analizy obrazu w zakresie detekcji dużych obiektów. Pomimo tego, że sformułowanie „*duże obiekty*” nie precyzuje bezpośrednio jakiego rodzaju mają to być obiekty, to we wstępnej części pracy (Rozdz. 1) wskazano cechy takich obiektów. Pozwalają one nam zaliczyć bądź wykluczyć obiekt na analizowanych obrazach do takiej grupy. Szczególną cechą, która była analizowana w tej pracy w największym stopniu, była niejednorodność dużych obiektów, która znacznie utrudnia korzystanie ze znanych algorytmów przetwarzania obrazu i często wymusza konstruowanie dedykowanych metod detekcji obiektów rozwiązujących ściśle określone zadanie. Metody te nie tylko, że mają wbudowaną wiedzę o analizowanym obiekcie od samego początku, to dodatkowo wymagają często ścisłej standaryzacji procesu akwizycji obrazu. W niniejszej pracy zaproponowano odejście od tego schematu, zalecając takie konstruowanie metod detekcji dużych obiektów, aby wiedzę o ich lokalizacji i innych cechach je wyróżniających mógł przekazać użytkownik na wstępnym etapie konfiguracji. Problematyce związanej z definiowaniem tej wiedzy w postaci modelu obiektu poświęcono III część pracy „[Algorytmy segmentacji oraz detekcji dużych obiektów](#)”. Zaproponowano tutaj 3 strategie określania dużych obiektów:

- Model bazujący na krawędzi (Rozdz. 6),
- Model bazujący na kształcie (Rozdz. 7),
- Model bazujący na znacznikach (Rozdz. 8).

Pierwszy z tych modeli ma zadanie zmniejszenia liczby segmentów w ramach dużych obiektów. Zadaniem drugiego modelu jest umożliwienie lokalizacji poszukiwanego obiektu na obrazie, z wykorzystaniem wzorca kształtu przygotowanego przez użytkownika. Ostatni model jest zalecany do interaktywnej segmentacji obrazu, gdzie użytkownik na bieżąco nadzoruje wyniki działania algorytmu. Pomimo tego, że modele te bazują na wielu znanych w literaturze algorytmach przetwarzania obrazu, takich jak algorytm działów wodnych, detekcja krawędzi algorytmem *Canny*, algorytmy nakładania obrazu oraz segmentacja bazująca na atlasie, to zawierają one również autorskie rozwiązania opracowane w trakcie

prac z obrazami przedstawiającymi duże obiekty. Do nowo wprowadzonych metod, które zaprezentowano w pracy można zaliczyć:

- algorytm łączenia segmentów uwzględniający wyniki wstępnej segmentacji obrazu algorytmem działów wodnych oraz położenie linii krawędzi *Canny* (Rozdz. 6.2),
- algorytm ścisłego dopasowania wzorcowego modelu do rzeczywistego obiektu na obrazie po etapie nakładania (Rozdz. 7.2).

Algorytmy te, chociaż nie stanowią samodzielnych metod detekcji dużych obiektów, to w znacznej mierze wpływają na powodzenie stosowania wskazanych w pracy modeli. Modele te, usprawnione własnymi algorytmami pozwalają w dużym stopniu na:

- łatwą re-konfigurację algorytmu detekcji dużych obiektów dla różnych grup obrazów przez użytkowników nie posiadających specjalistycznej wiedzy w zakresie komputerowej analizy obrazu,
- pełną automatyzację procesu detekcji, po zakończeniu wstępnego etapu konfiguracji,
- łatwą i powtarzalną metodę korekcji ewentualnych błędów detekcji z wykorzystaniem metod interaktywnych.

W celu zwiększenia skuteczności działania proponowanych metod detekcji dużych obiektów, w II części pracy omówiono sposoby przygotowania obrazu do analizy. Wskazano tutaj głównie na klasyczne i zaawansowane techniki usuwania szumu z obrazu oraz na metody wstępnej segmentacji obrazu. Oprócz tego, w dwóch dodatkach (A i B) do niniejszej pracy opisano istotne algorytmy stanowiące bazę dla proponowanych metod, aby dokładniej wyjaśnić ich działanie czytelnikom, którzy nie zajmują się zawodowo przetwarzaniem obrazu.

Praktyczne zastosowanie omawianych w pracy metod i modeli zademonstrowano poprzez:

- analizę serii przykładowych obrazów, ze szczególnym uwzględnieniem obrazów medycznych
- oraz przygotowanie przykładowego systemu graficznej bazy danych, z wdrożonymi metodami, pracującej w trybie automatycznym.

### Wnioski

1. Detekcja dużych obiektów na obrazach o wysokim poziomie szumu i niejednorodności nie musi odbywać się z wykorzystaniem metod i algorytmów dedykowanych do każdego typu obrazu osobno. Możliwe jest opracowanie bardziej uniwersalnych metod i narzędzi, które korzystając z wiedzy i doświadczenia użytkownika, mogą być stosowane do różnych serii zdjęć. Zaproponowane w pracy trzy strategie określania dużych obiektów pozwalają tworzyć modele reprezentujące wybrane cechy tych obiektów, które następnie mogą być wykorzystane do ich automatycznej detekcji.
2. Wykorzystanie zaproponowanych modeli opisywania dużych obiektów pozwala na konstruowanie elastycznych narzędzi przetwarzania obrazu, które po etapie wstępnej konfiguracji mogą być stosowane dla różnych serii obrazów. W pracy pokazano, że taka konfiguracja nie musi wymagać od użytkownika specjalistycznej wiedzy w zakresie konstrukcji algorytmów przetwarzania obrazu, a może polegać na manualnym zakreśleniu jednego lub kilku obiektów na wybranych obrazach z całej serii zdjęć.
3. Proponowana w niniejszej pracy metoda detekcji bazująca na modelu kształtu, pozwala na pełną automatyzację procesu detekcji, po zakończonym etapie konfiguracji.
4. Zastosowanie znanych metod nakładania obrazu do detekcji obiektów daje jedynie przybliżone rezultaty. W szczególności problem ten dotyczy dokładnego wyznaczenia granic obiektów. Wykorzystanie zaproponowanego w niniejszej pracy algorytmu ścisłego dopasowania wzorcowego modelu do obiektu poprawia znacznie rezultaty tej operacji.
5. Wskazane w niniejszej pracy ograniczenia stosowania metody bazującej na wzorcowym kształcie wpływają na zakres ich stosowania w diagnostyce obrazowej. Wymóg podobieństwa w zakresie kształtu, rozmiaru i położenia analizowanych obiektów względem obiektu wzorcowego uniemożliwia stosowanie ich do obrazów tkanek miękkich, natomiast pozwala na użycie ich w przypadku analizy obrazów układu kostnego.
6. Tradycyjne metody segmentacji obrazu często dzielą duże obiekty na wiele mniejszych regionów. Standardowo zalecane w takiej sytuacji stosowanie narzędzi łączących te regiony, w przypadku obiektów niejednorodnych nie daje pożądanego rezultatu (problem określenia kryterium spójności pikseli).

Proponowany w niniejszej pracy algorytm łączący segmenty z wykorzystaniem linii krawędzi Canny, rozwiązuje częściowo ten problem. Warunkiem skuteczności jego działania jest istnienie widocznej granicy między dwoma analizowanymi obiektami, lub między obiektem a jego tłem.

7. Z powodu przyjętych założeń o ogólnym charakterze proponowanych metod i próbie ich pełnej automatyzacji, trudno zagwarantować skuteczność ich działania w każdym przypadku. W związku z tym, przydatność ich użycia powinna być każdorazowo oceniana przez użytkownika dla nowej serii zdjęć, i jeśli błąd detekcji będzie zbyt duży, należy wówczas rozważyć użycie opisywanej w niniejszej pracy metody interaktywnej segmentacji dużych obiektów lub też konstrukcję dedykowanego algorytmu rozwiązującego wcześniej zdefiniowany problem.

**Część V**

**Dodatki**



*Algorytmy detekcji krawędzi*

**A.1 Proste sposoby detekcji krawędzi**

Najczęściej używaną metodą detekcji krawędzi jest zastosowanie operatora gradientu. Jego wyliczenie dla obrazu cyfrowego nie jest skomplikowane i w uproszczeniu sprowadza się do znalezienia różnicy między sąsiednimi punktami. W zależności od sposobu wyboru tych punktów możemy skorzystać z jednego z popularnych filtrów gradientowych, a mianowicie: *Roberts*, *Prewitt* lub *Sobel* (odpowiednie macierze filtrów przedstawiono w tabeli A.1). Podane filtry działają w dwóch kierunkach: poziomym  $G_x$  i pionowym  $G_y$ , dlatego wyznaczenie wartości odpowiedzi  $G_{mag}$  i kierunku gradientu  $G_{dir}$  wymaga dodatkowych obliczeń. W tym celu korzystamy ze wzorów:

$$G_{mag} = \sqrt{G_x^2 + G_y^2} \tag{A.1}$$

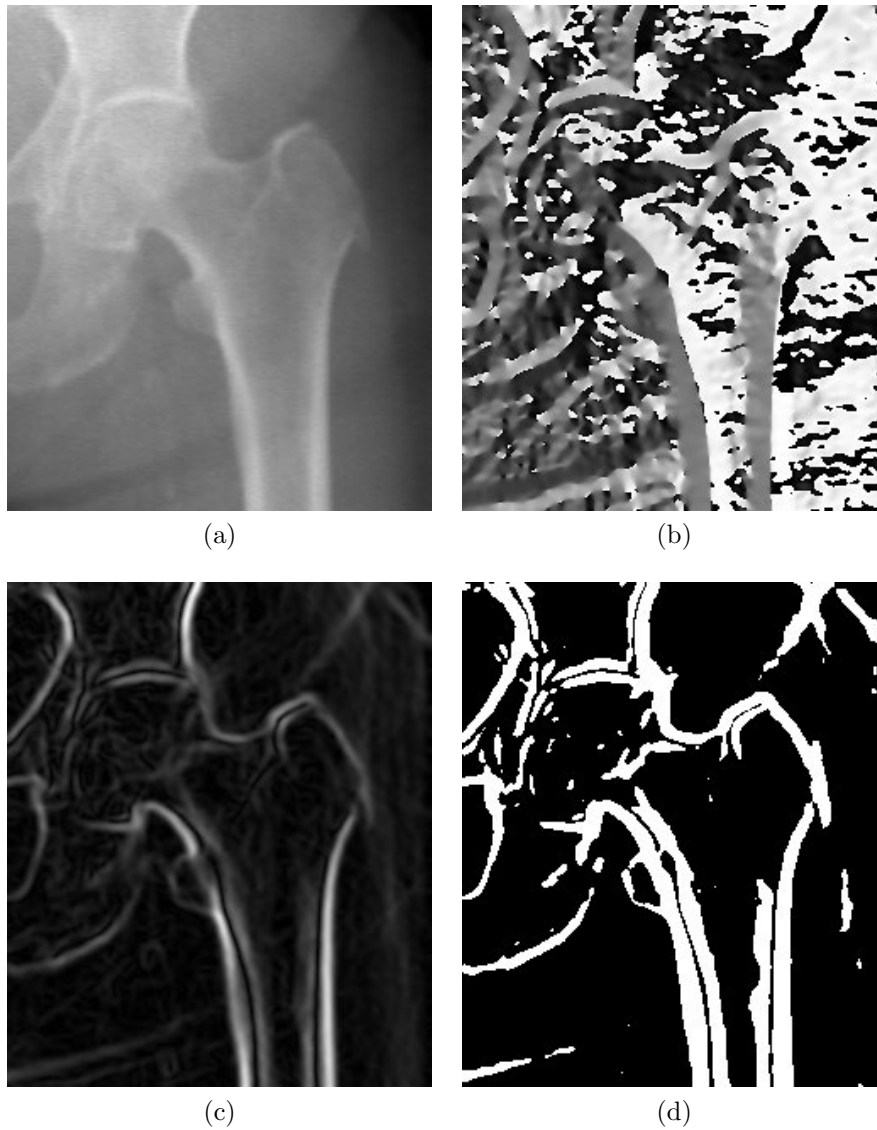
$$G_{dir} = \arctan G_y/G_x \tag{A.2}$$

Na obrazie, który poddano działaniu filtru gradientowego (Rys. A.1c) najjaśniejsze punkty pokrywają się z krawędzią obrazu. Wynika to z faktu, że w tych miejscach zmiana intensywności sąsiednich pikseli jest największa. Aby jednoznacznie wskazać, które punkty należą do krawędzi, a które nie, należy wybrać wartość progową i przeprowadzić operację binaryzacji. Kluczowe w tym wypadku właściwe określenie progu, często musi zostać zrealizowane na zasadzie kompromisu – jeśli zostanie wybrana zbyt niska wartość,

Tabela A.1: Macierze filtrów gradientowych w kierunku poziomym  $G_x$  i pionowym  $G_y$ .

-1	0	+1	-1	-1	-1	+1	0	-1	+1	+2	+1
-1	0	+1	0	0	0	+2	0	-2	0	0	0
-1	0	+1	+1	+1	+1	+1	0	-1	-1	-2	-1
(a) Prewitt $G_x$			(b) Prewitt $G_y$			(c) Sobel $G_x$			(d) Sobel $G_y$		
+1		0		0		0		+1			
0		-1		-1		-1		0			
(e) Roberts $G_x$						(f) Roberts $G_y$					

to zaznaczony będzie również szum, ewentualnie drobne krawędzie i rysy przecinające właściwe obiekty. Z drugiej strony, ustawienie zbyt dużej wartości wprowadzi podziały w krawędziach i znacznie utrudni dalszą segmentację.



Rysunek A.1: Detekcja krawędzi z użyciem obrazu gradientowego przygotowanego filtrem *Sobela*: (a) obraz oryginalny, (b) kierunek gradientu, (c) wartość gradientu, (d) binaryzacja z progiem o wartości 38 (*opracowanie własne*).

## A.2 Zaawansowana detekcja krawędzi na przykładzie *algorytmu Canny*

Istnieje wiele zaawansowanych algorytmów detekcji krawędzi, które znacznie niwelują problemy prostych filtrów gradientowych opisanych w poprzednim podrozdziale. Możemy do nich zaliczyć m.in. algorytmy [40, 57, 15]. W dalszej części tekstu zostanie szerzej opisany *algorytm Canny* [7], który użyto w metodzie zaprezentowanej w rozdziale 6.2.

John F. Canny w 1986 roku zaproponował [7] kilkietapowe podejście do detekcji krawędzi, spełniające zadane kryteria:

- kryterium dobrej detekcji - algorytm powinien zaznaczyć tyle rzeczywistych krawędzi obrazu, ile tylko jest możliwe;
- kryterium dobrej lokalizacji - zaznaczone krawędzie powinny być możliwie blisko krawędzi obrazu rzeczywistego;
- kryterium pojedynczej odpowiedzi - właściwe krawędzie obrazu powinny zostać zaznaczone tylko raz, a tam gdzie to możliwe zakłócenia obrazu nie powinny tworzyć fałszywych krawędzi.

Przedstawił on również praktyczne rozwiązanie tego podejścia opierające się na sekwencji następujących operacji:

### 1. Redukcja szumu

W celu rozmycia obrazu, i tym samym usunięcia lokalnych zakłóceń, autor algorytmu zaproponował użycie filtru bazującego na pierwszej pochodnej funkcji Gaussa (szerzej opisane w rozdziale 3.2) (Rys. A.2).

### 2. Detekcja krawędzi

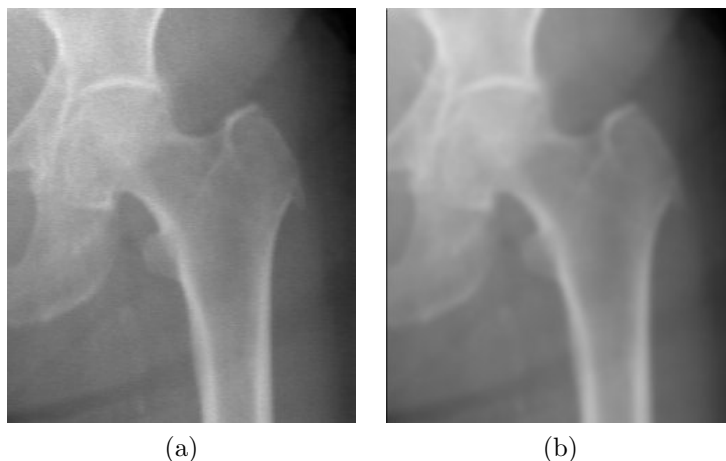
Ta operacja może zostać przeprowadzona za pomocą wybranego operatora gradientu, które opisano w poprzednim podrozdziale. Na potrzeby kolejnego etapu Canny proponuje zaokrąglenie macierzy kierunków gradientu  $G_{dir}$  do  $45^\circ$  (Rys. A.3).

### 3. Eliminacja pikseli o niemaksymalnej wartości<sup>1</sup>

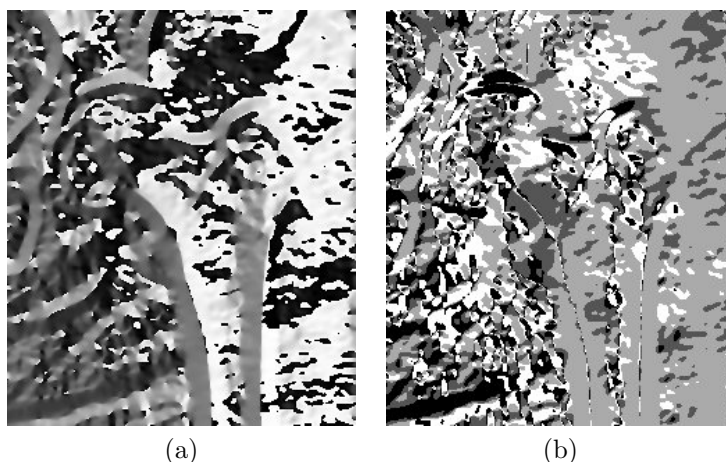
Aby spełnić postawione *kryterium dobrej lokalizacji*, został wprowadzony algorytm usuwania pikseli, których wartość jest niższa od wartości sąsiadów w kierunku wskazanym przez macierz  $G_{dir}$ . Po tej operacji na obrazie zostaną tylko najsilniejsze punkty poszczególnych krawędzi, które powinny znajdować się blisko rzeczywistej granicy obiektów (Rys. A.4).

---

<sup>1</sup>Oryginalna, angielska nazwa wprowadzona przez autora to *Non-maximum suppression*.



Rysunek A.2: Redukcja szumu filtrem Gaussa o wartości  $\sigma = 2.0$ .  
(a) Obraz oryginalny. (b) Obraz wynikowy (*opracowanie własne*).



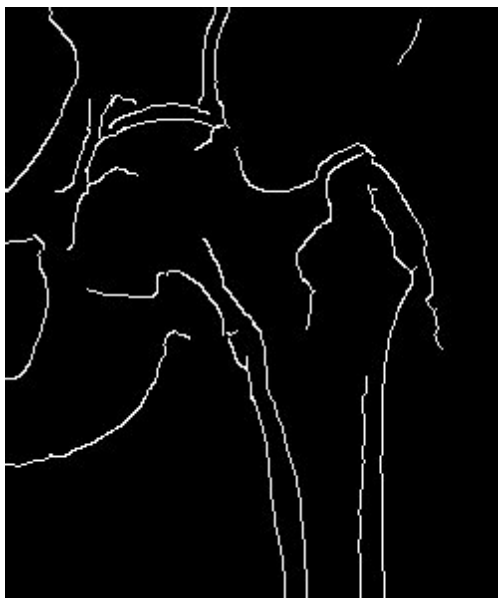
Rysunek A.3: (a) Kierunek gradientu i (b) jego zwrot (wartości dla przeciwległych kątów są takie same) z dokładnością do  $45^\circ$  (*opracowanie własne*).

#### 4. Konstrukcja krawędzi

W ostatnim etapie zaleca się przeprowadzenie progowania z histerezą. Różni się ono tym od tradycyjnego progowania, że operacja binaryzacji jest wykonywana dwukrotnie — z użyciem progu wysokiego  $T_h$  i progu niskiego  $T_l$ , a wyniki zapisywane są na osobnych obrazach. Ostatecznie uwzględniamy tylko te punkty obrazu z niskim progiem, które łączą się z punktami obrazu wysokiego progu (może to zostać zrealizowane z użyciem technik śledzenia konturu (ang. *edge tracking*) lub operacji morfologicznej zwanej rekonstrukcją przez dylatację). Działając w ten sposób można zachować kompromis między skutecznością wykrycia krawędzi, a eliminacją szumów (Rys. A.5).



Rysunek A.4: Wynik eliminacji pikseli o niemaksymalnej wartości (*opracowanie własne*).



Rysunek A.5: Obraz wynikowy dla algorytmu detekcji krawędzi *algorytmem Canny* [7] (*opracowanie własne*).

Opisany algorytm w porównaniu do metody przedstawionej w podrozdziale [A.1](#), nie tylko pozwala na uzyskanie cienkich linii krawędzi zaraz po jego zakończeniu (Rys. [A.5](#)), ale dodatkowo uzyskane linie są bardziej wydłużone i nie zawierają takiej ilości szumu, jak to ma miejsce w przypadku binaryzacji gradientu globalną wartością progową (Rys. [A.1d](#) na stronie [116](#)).

### *Algorytmy wyznaczające transformację jednego obrazu w drugi*

W wielu problemach przetwarzania obrazu występuje potrzeba analizy zdjęć tych samych obiektów, których akwizycja została wykonana za pomocą różnych urządzeń lub w różnym odstępie czasu. Zapotrzebowanie na tak zróżnicowane pomiary może wynikać z faktu niewystarczającej ilości informacji pozyskanej jednym urządzeniem do rejestracji obrazu, lub też wtedy, kiedy obiekt badań zmienia swoje położenie lub kształt w czasie. To, co dla człowieka jest naturalne podczas analizy takich zdjęć, a mianowicie fakt, że ma do czynienia z tym samym obiektem widzianym z różnej perspektywy, nie jest już oczywiste dla komputera. W najprostszym przypadku maszyna jest w stanie stwierdzić, że ma do czynienia z tym samym obiektem badań, jeśli wszystkie jego piksele na poszczególnych obrazach mają taką samą wartość, lub ta wartość zmienia się proporcjonalnie (np. z powodu zmiany oświetlenia). Natomiast kiedy obiekt znajduje się w różnych miejscach obrazu, a tym bardziej ma zmieniony kształt, może okazać się celowe zastosowanie zaawansowanych algorytmów nakładania obrazów, które umożliwiają wyznaczenie wektora transformacji jednego obrazu w drugi. Algorytmy tego typu znajdują szczególne zastosowanie w następującej klasie problemów:

- Techniki śledzenia obiektów na obrazach rejestrowanych za pomocą kamery.  
Obrazy pozyskane za pomocą kamer mogłyby okazać się bezwartościowe z punktu widzenia ich cyfrowego przetwarzania, gdyby nie techniki analizy ruchu. Techniki te umożliwiają śledzenie wybranych obiektów i badanie jaki wpływ miało otoczenie na ich ruch. Znajdują one ważne zastosowanie w systemach monitoringu, sterowania oraz kontroli jakości.
- Aktualizacja zdjęć lotniczych i satelitarnych.  
Ilość wykonywanych zdjęć terenu - zarówno przez satelity jak i specjalistyczne samoloty - przyrasta w bardzo szybkim tempie. Do ich cyfrowej obróbki nie wystarczają informacje geolokacyjne, ale również wykorzystywane są specjalistyczne algorytmy łączące te zdjęcia oraz dopasowujące do wcześniej zgromadzonych w kartograficznej bazie danych.



- Diagnostyka obrazowa.

Potrzeba stałej kontroli stanu zdrowia pacjenta, szczególnie w wypadku postępującej choroby lub też monitoringu pooperacyjnego, może wymagać okresowych badań radiologicznych. Ich skuteczna analiza za pomocą systemów komputerowych może wymagać wcześniejszego zestawienia odpowiadających sobie organów i dopiero późniejszą ocenę występujących zmian. Podobna sytuacja ma miejsce w przypadku badania tych samych obiektów za pomocą różnych urządzeń akwizycyjnych (np. łączenie informacji z pozytonowej emisyjnej tomografii komputerowej - *PET* i tomografii komputerowej - *CT* podczas analizy zmian w mózgu).

Pomimo tego, że opisane przykłady mają wiele cech wspólnych - min. dotyczą analizy tych samych obiektów na wielu obrazach - to mają one również cechy różniące je między sobą:

- Analizowane obiekty na zdjęciach z kamery często zmieniają swoje położenie względem siebie, jednak ta zmiana na poszczególnych klatkach filmu nie jest wielka.
- Na zdjęciach satelitarnych obiekty rzadko zmieniają położenie względem siebie. Są one natomiast fotografowane z różnej perspektywy i mogą znajdować się w zupełnie innych miejscach na kolejnych obrazach.
- Na obrazach radiologicznych obiekty badań często mają zmieniony kształt, a rzadko położenie. Zazwyczaj zestawiane są dwa obrazy ze sobą do porównania.

Z powodu różnej specyfiki przedstawionych przykładów zostały wypracowane różne algorytmy sprawdzające się najlepiej w konkretnym zastosowaniu. Zostaną one omówione w kolejnych podrozdziałach z wykorzystaniem obrazów z rysunku 7.1 (strona 69).

## B.1 Techniki śledzenia obiektów

Dziedzina przetwarzania obrazów związana z analizą ruchu dostarcza szeregu algorytmów, które mogą być z powodzeniem stosowane do śledzenia tego, w jaki sposób badany obiekt zmienia swoją lokalizację (ewentualnie kształt) na kolejnych obrazach w całej serii zdjęć<sup>1</sup>. Algorytmy tego typu możemy podzielić na dwie grupy<sup>2</sup>:

- algorytmy, które pozwalają jedynie na wykrycie samego faktu ruchu i ewentualnie wskazują obszar obrazu na którym nastąpiła zmiana (zazwyczaj za pomocą prostokątnego obszaru),

---

<sup>1</sup>Zazwyczaj analizie podlegają poszczególne klatki filmu wideo lub obrazu rejestrowanego kamerą

<sup>2</sup>Jest to bardzo uproszczony podział algorytmów śledzenia obiektów, który w zupełności wystarcza na potrzeby tej pracy.

- algorytmy, które dodatkowo umożliwiają obliczenie wektora ruchu pomiędzy dwoma obrazami w sekwencji występującymi po sobie.

Interesujące z punktu widzenia niniejszej pracy algorytmy z grupy drugiej określane są mianem algorytmów przepływu optycznego (ang. *Optical flow*). Umożliwiają one wyznaczenie wektora ruchu, który pozwala na przekształcenie danego obrazu w sekwencji zdjęć w kolejny obraz w tej sekwencji. Przemieszczane są w tym celu obszary z pierwszego obrazu, zgodnie z odpowiadającymi im wektorami w drugi obraz.

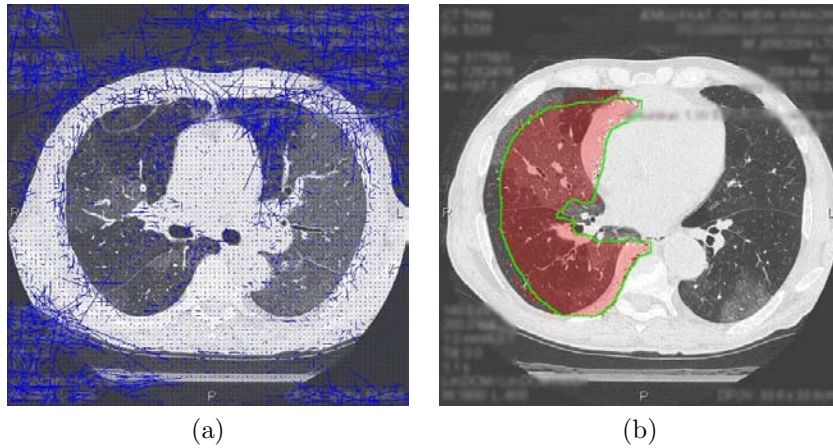
Istnieje wiele algorytmów przepływu optycznego, które znacznie różnią się sposobem działania. W uproszczeniu można je podzielić na trzy grupy:

- metody gradientowe - bazują na analizie pochodnych intensywności obrazu;
- metody częstotliwościowe - oparte są na filtrowaniu informacji obrazowej w dziedzinie częstotliwości;
- metody korelacyjne - bazujące na odpowiedniości obszarów obrazu.

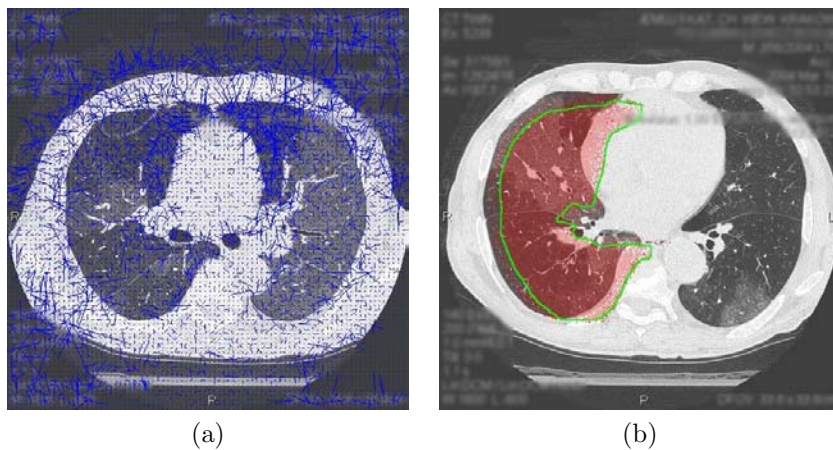
Na potrzeby tej pracy przetestowano *algorytm Lucas-Kanade* [38] i *algorytm Horn-Schunck* [27] zaliczane do algorytmów gradientowych oraz *algorytm korelacyjny dopasowania bloków* [78] (ang. *Block matching*). Wyniki testów w postaci graficznej pokazano na rysunkach B.1, B.2 i B.3. Przedstawiają one obraz 7.1a, na który naniesiono pole przepływu optycznego (niebieskie linie na obrazie w pierwszej kolumnie) wyliczone podczas nakładania na obraz 7.1b. Następnie wykonywana jest transformacja wcześniej przygotowanego obrysu obiektu (zielona linia na obrazie w drugiej kolumnie) zgodnie z wyliczonym polem przepływu. Wynikowy obiekt jest naniesiony na obrazie 7.1b w postaci czerwonego obszaru.

Niestety, żaden z analizowanych algorytmów przepływu optycznego nie dał satysfakcjonujących efektów. Problemem może się okazać brak spełnionych założeń wstępnych do używania testowanych algorytmów:

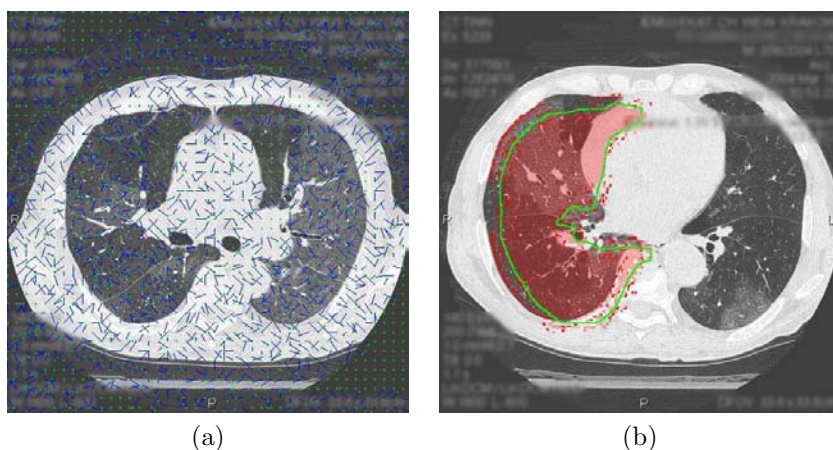
- Algorytmy gradientowe najlepiej sprawdzają się tam gdzie prędkość poruszania się obiektów nie przekracza jednego punktu na ramkę obrazu, w przeciwnym wypadku wyliczenie odpowiednich pochodnych jest albo niemożliwe, albo obciążone dużym błędem.
- W przypadku metod korelujących fragmenty obrazu proces korelacji opiera się na założeniu, że jeśli sąsiednie punkty należą do tego samego obiektu, to poruszają się one w tym samym kierunku. Założenie to jest słuszne, gdy rozpatrywany jest ruch ciał sztywnych, a transformacja ogranicza się do przemieszczenia.



Rysunek B.1: Prezentacja działania algorytmu przepływu optycznego *Lukas-Kanade* [38] (opracowanie własne).



Rysunek B.2: Prezentacja działania algorytmu przepływu optycznego *Horn-Schunck* [27] (opracowanie własne).



Rysunek B.3: Prezentacja działania korelacyjnego algorytmu dopasowania bloków [78] (opracowanie własne).

## B.2 Algorytmy nakładania obrazów

W odróżnieniu od technik śledzenia obiektów metody nakładania obrazów opisane w tym podrozdziale zazwyczaj operują na dwóch obrazach wejściowych, z których jeden jest obrazem odniesienia, a drugi jest obrazem nakładanym. Celem algorytmów nakładania nie jest jedynie określenie wektora ruchu – jak to miało miejsce w algorytmach przepływu optycznego, ale również transformacja obrazu nakładanego na obraz odniesienia. Na przestrzeni ostatnich lat powstało wiele algorytmów tego typu. Próba ich klasyfikacji została podjęta w opracowaniach [39, 79]. Na potrzeby niniejszej pracy zostanie przyjęty jedynie uproszczony podział tych metod ze szczególnym wskazaniem na te, które mogą okazać się przydatne dla testowanych obrazów płuc (Rys. 7.1) i innych, na których testowano proponowaną metodę (Rozdz. 7.2).

Algorytmy nakładania obrazów ze względu na rodzaj przyjętej transformacji podczas poszukiwania optymalnego dopasowania można podzielić na dwie grupy:

- algorytmy globalne, które zachowują wzajemne położenie poszczególnych pikseli względem siebie – pozwalają one na stosowanie liniowych przekształceń, zwanych również przekształceniami sztywnymi (ang. *rigid registration*), do których można zaliczyć operacje obracania, skalowania i inne,
- algorytmy lokalne – umożliwiają one deformację (ang. *non-rigid registration*, *warping registration*, *deformable registration*) obrazu nakładanego, aby umożliwić lepsze dopasowanie poszczególnych obiektów.

Algorytmy globalne, które nie przemieszczają pikseli względem siebie – przez co działają szybciej – zaleca się stosować tam, gdzie położenie poszczególnych obiektów względem siebie nie uległo zmianie, a jedynie zmieniła się perspektywa obserwacji tych obiektów. Taka sytuacja zazwyczaj ma miejsce, jeśli akwizycja obiektów odbywała się z użyciem innych urządzeń, lub obiekt był obserwowany z różnych miejsc, więc zmienił się kąt widzenia. Natomiast algorytmy lokalne najlepiej używać, jeśli dodatkowo zmianie uległ kształt/forma badanego obiektu.

Niezależnie od tego, czy algorytm będzie nakładał obrazy stosując podejście lokalne czy globalne, przygotowując jego implementację można bazować na wcześniej wyznaczonych cechach charakterystycznych obrazu (ang. *feature-based image registration*) lub też na wartościach intensywności wszystkich pikseli (ang. *intensity-based image registration*). Wśród cech charakterystycznych pomocnych podczas nakładania obrazów można wyróżnić punkty i linie zaznaczone przez użytkownika lub naniesione automatycznie przez dodatkowo skonstruowany algorytm. Przydatny może być również wynik detekcji krawędzi lub wstępna detekcja obiektów na obu obrazach. Natomiast w podejściu bazującym

na intensywności pikseli korzysta się z pełnej informacji o obrazie. W tym podejściu możliwe jest uzyskanie większej dokładności nałożenia obrazów, ale odbywa się to kosztem zwiększonego zapotrzebowania na moc obliczeniową.

W dalszej części rozdziału zostaną opisane lokalne algorytmy nakładania obrazów bazujące na intensywności punktów, ponieważ mają one bardziej ogólny charakter i pozwalają na osiągnięcie lepszej dokładności. Wybór tej grupy algorytmów jest również podyktowany naturą testowych obrazów, na których badany obiekt zmienia swój kształt oraz w nieznacznym stopniu położenie.

### Lokalne algorytmy nakładania obrazów

W celu dobrania odpowiedniego algorytmu nakładania obrazu, a zarazem skrócenia etapu jego poszukiwania, zawężono zakres algorytmów do implementacji zawartych w narzędziach *Insight Toolkit* [84] i *ImageJ* [82]<sup>3</sup>. Wymienione narzędzia posiadają największy zestaw zaimplementowanych algorytmów nakładania obrazów wśród narzędzi znanych autorowi pracy i dostępnych za darmo razem z kodem źródłowym. Dla każdego z tych narzędzi przetestowano wszystkie dostępne w nich algorytmy i najlepsze wyniki działania zostaną pokrótce omówione i pokazane na rysunkach.

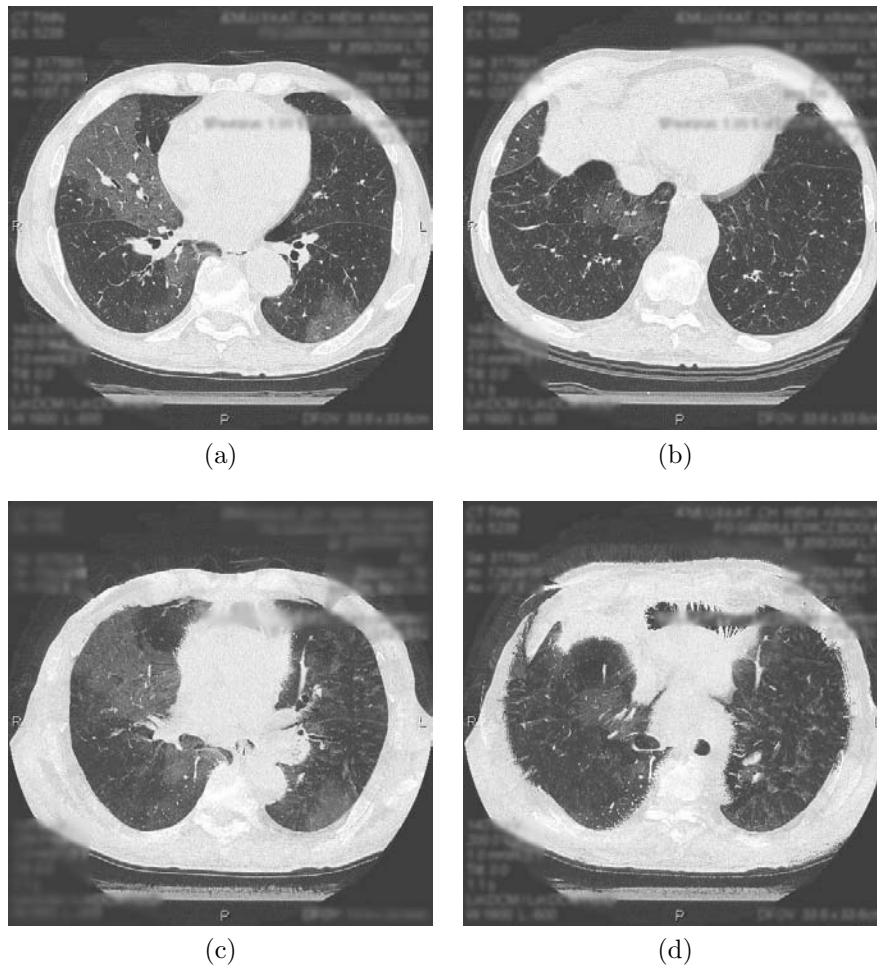
- W *Insight Toolkit* najlepsze rezultaty udało się uzyskać z wykorzystaniem algorytmu bazującego na ruchu poziomim (ang. *Level set motion*) przedstawionego w pracy [72]<sup>4</sup>. Jak można zobaczyć na rysunku B.4 próba dopasowania obrazu 7.1a do obrazu 7.1b (został on powtórnie pokazany na rysunku B.4a w celu łatwiejszej analizy wzrokowej wyników) zakończyła się satysfakcjonującym wynikiem (Rys. B.4c), czego nie można już powiedzieć o drugim przykładzie, w którym obrazem odniesienia był obraz 7.1c (pokazany ponownie jako B.4b). Wykonana deformacja zupełnie zniekształciła obraz nakładany 7.1a, przez co metoda ta nie może zostać uznana za użyteczną dla analizowanych obrazów testowych (Rys. B.4d).
- Z kolei w *ImageJ* najlepsze rezultaty udało się osiągnąć przy użyciu dodatku *bUnwarpJ* [80]. Zaimplementowany tam został algorytm zaproponowany w pracy [3]. Łączy on w sobie najlepsze cechy znanych wcześniej metod nakładania obrazów [67] i [10], gdzie pole deformacji jest modelowane z użyciem krzywych B-Spline, a samo nakładanie może być liczone dla obu kierunków na raz (z obrazu nakładanego do obrazu odniesienia i na odwrót). Przykład przedstawiony na rysunku B.5 pokazuje

---

<sup>3</sup>To uproszczenie jest podyktowane tym, że algorytmy nakładania obrazu nie stanowią właściwej części tej pracy. Podjęto się jednak przedstawienia podstawowych zasad ich działania, aby przybliżyć kontekst działania proponowanego algorytmu, który zostanie opisany w dalszej części tego rozdziału.

<sup>4</sup>Skorzystano z funkcji `LevelSetMotionRegistrationFilter`, której szczegółowy opis jest dostępny na stronie internetowej [85].



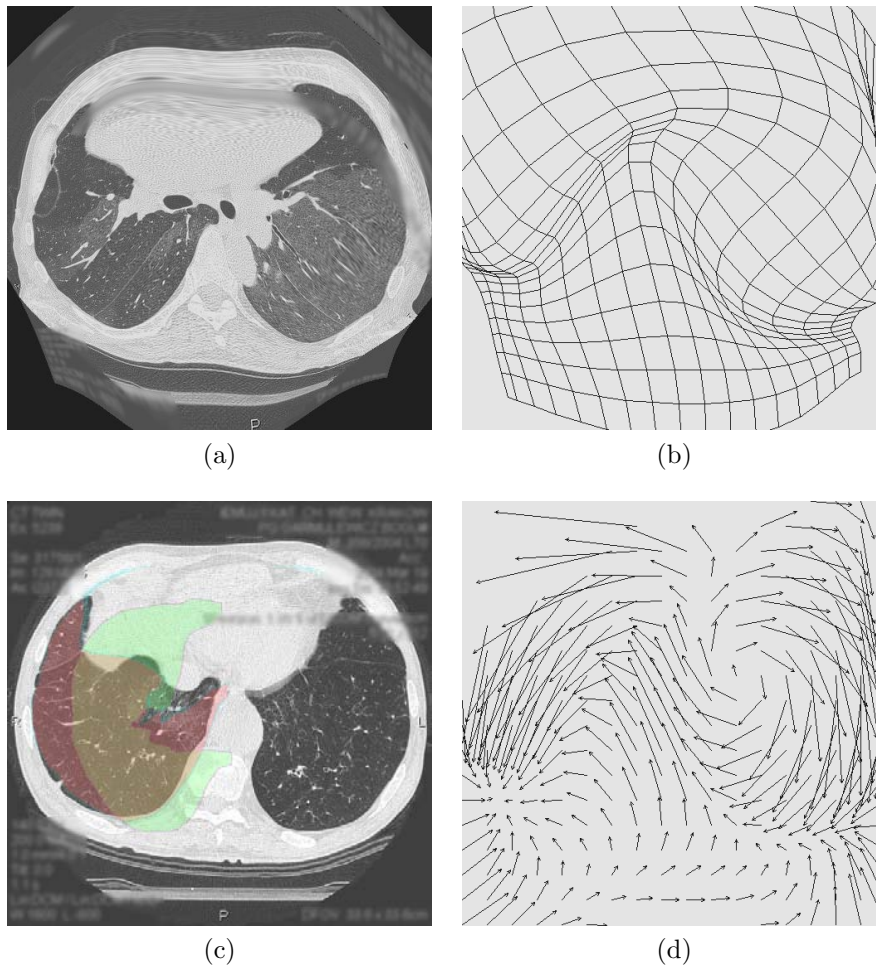


Rysunek B.4: Prezentacja działania algorytmu nakładania obrazów bazującego na ruchu poziomym zaimplementowanego w bibliotece *Insight Toolkit* [84] (opis w tekście, opracowanie własne).

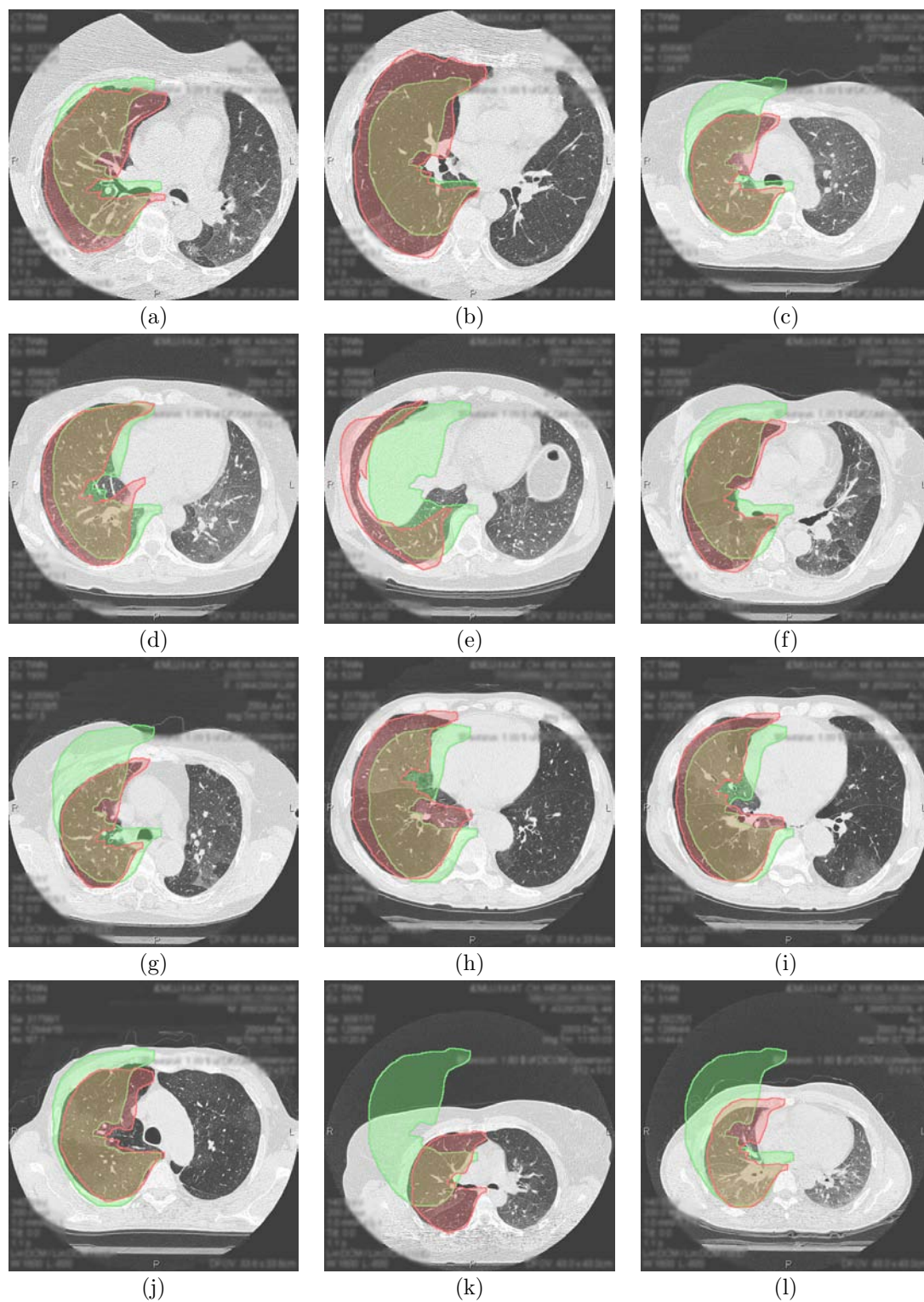
transformację obrazu 7.1a w obraz 7.1c<sup>5</sup>. Obraz B.5a przedstawia obraz nakładany po zakończonej deformacji. Siatka deformacji jest widoczna obok na obrazie B.5b. Poniżej na obrazie B.5d w sposób graficzny pokazano pole wektorowe transformacji punktów. Zostało ono użyte do transformacji obrazu z odręcznie zaznaczonym prawym płucem (Rys. 7.2), które następnie naniesiono na obraz odniesienia B.5c jako czerwony obszar. Dodatkowo na obrazie B.5c kolorem zielonym zaznaczono wstępny obszar płuc.

<sup>5</sup>Przykład dla obrazu 7.1b jak i innych obrazów odniesienia zostanie pokazany na rysunku B.6.





Rysunek B.5: Prezentacja działania dwukierunkowego algorytmu nakładania obrazów z wykorzystaniem krzywych B-Spline zaimplementowanego w narzędziu *bUnwarpJ* [80] (*opis w tekście, opracowanie własne*).



Rysunek B.6: Rezultaty nakładania obrazu uzyskane narzędziem *bUnwarpJ* [80]. Kolor zielony - obiekt wzorcowy. Kolor czerwony - obiekt zdeformowany (*opracowanie własne*).

---

## Literatura

- [1] R. Adams, L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:641–647, 1994. (Cytowano na stronach: 16)
- [2] A. A. Amini, T. E. Weymouth, R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990. (Cytowano na stronach: 16 i 17)
- [3] Ignacio Arganda-Carreras, Carlos Oscar Sánchez Sorzano, Roberto Marabini, José María Carazo, Carlos Ortiz de Solorzano, Jan Kybic. Consistent and elastic registration of histological sections using vector-spline regularization. R. Beichel, M. Sonka, redaktorzy, *International Workshop on Computer Vision Approaches to Medical Image Analysis*, wolumen 4241 serii *Lecture Notes in Computer Science*, strony 85–95. Springer, 2006. (Cytowano na stronach: 102 i 125)
- [4] A. Bleau, L. J. Leon. Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, 77(3):317–370, 2000. (Cytowano na stronach: 16, 17 i 45)
- [5] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, May 2009. (Cytowano na stronach: 18)
- [6] M. Buvry, J. Senard, C. Krey. Hierarchical region detection based on gradient image. *10th Scandinavian Conference on Image Analysis*, wolumen 2, strony 717–724, 1997. (Cytowano na stronach: 56)
- [7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986. (Cytowano na stronach: 15, 48, 49, 117 i 119)
- [8] F. Catté, P. L. Lions, J. M. Morel, T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 29(1):182–193, 1992. (Cytowano na stronach: 30)
- [9] L. Chmielewski. Choice of the hough transform for image registration. wolumen 5505, strony 122–134. SPIE, 2004. (Cytowano na stronach: 68)

- [10] G. E. Christensen, J. He. Consistent nonlinear elastic image registration. *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, 0:37 – 43, 2001. (Cytowano na stronach: 125)
- [11] R. Clouard, A. Elmoataz, Ch. Porquet, M. Revenu. Borg: A knowledge-based system for automatic generation of image processing programs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2):128–144, 1999. (Cytowano na stronach: 18)
- [12] T. Cootes. Active shape Models-Their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995. (Cytowano na stronach: 18)
- [13] T. F. Cootes, C. J. Taylor. Active shape models - „smart snakes”. *Proceedings of the British Machine Vision Conference*, 1992. (Cytowano na stronach: 18)
- [14] Y. Deng, B. S. Manjunath, C. S. Kenney, M. S. Moore, H. Shin. An efficient color representation for image retrieval. *IEEE Transactions on Image Processing*, 10(1):140–147, 2001. (Cytowano na stronach: 105)
- [15] P. Dollár, Z. Tu, S. Belongie. Supervised learning of edges and object boundaries. *IEEE Conference on Computer Vision and Pattern Recognition*, strony 1964–1971. IEEE Computer Society, 2006. (Cytowano na stronach: 117)
- [16] V. Duay, N. Houhou, J. P. Thiran. Atlas-based segmentation of medical images locally constrained by level sets. wolumen 2, strony II–1286–9, November 2005. (Cytowano na stronach: 72)
- [17] E. A. El-Kwae, M. R. Kabuka. A robust framework for content-based retrieval by spatial similarity in image databases. *ACM Transactions on Information Systems*, 17(2):174–198, 1999. (Cytowano na stronach: 105)
- [18] P. F. Felzenszwalb, D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004. (Cytowano na stronach: 17, 36, 37 i 38)
- [19] J. M. Fitzpatrick, D. L. G. Hill, C. R. Maurer. Image Registration. M. Sonka, J. M. Fitzpatrick, redaktorzy, *Handbook of Medical Imaging, Vol.2: Medical Image Processing and Analysis*, rozdział 8. SPIE Press, Lipiec 2000. (Cytowano na stronach: 68)
- [20] E.C. Freuder. Affinity: A relative approach to region finding. *Computer Graphics Image Processing*, 5(2):254–264, June 1976. (Cytowano na stronach: 16)

- [21] G. Gerig, O. Kübler, R. Kikinis, F.A. Jolesz. Nonlinear anisotropic filtering of mri data. 11(2):221–231, 11 1992. (Cytowano na stronach: 48)
- [22] O. Gómez, J. González, E. Morales. Image segmentation using automatic seeded region growing and instance-based learning. Luis Rueda, Domingo Mery, Josef Kittler, redaktorzy, *Progress in Pattern Recognition, Image Analysis and Applications*, wolumen 4756 serii *Lecture Notes in Computer Science*, strony 192–201. Springer Berlin / Heidelberg, 2007. (Cytowano na stronach: 16)
- [23] V. Grau, A. U. J. Mewes, M. Alcaniz, R. Kikinis, S. K. Warfield. Improved watershed transform for medical image segmentation using prior information. *IEEE Transactions on Medical Imaging*, 23(4):447–458, April 2004. (Cytowano na stronach: 72)
- [24] A. Gądek. *Komputerowa analiza obrazu regeneratu kostnego w metodzie Ilizarowa*. Rozprawa doktorska, Politechnika Krakowska, Kraków, 2005. (Cytowano na stronach: 4)
- [25] J. Hajnal, D. J. Hawkes, D. Hill. *Medical Image Registration*. CRC Press, 2001. (Cytowano na stronach: 68)
- [26] K. Haris, S. N. Efstratiadis, N. Maglaveras. Watershed-based image segmentation with fast region merging. *International Conference on Image Processing*, wolumen 3, strony 338–342, 1998. (Cytowano na stronach: 56)
- [27] B. K. P. Horn, B. G. Schunck. Determining optical flow. *Artificial Intelligence.*, 17(1-3):185–203, 1981. (Cytowano na stronach: 122 i 123)
- [28] S.L. Horowitz, T. Pavlidis. Picture segmentation by a directed split and merge procedure. *Proceedings International Association for Pattern Recognition (ICPR1974)*, strony 424–433, 1974. (Cytowano na stronach: 17)
- [29] D. N. F. A. Iskandar, J. A. Thom, S. M. M. Tahaghoghi. Content-based image retrieval using image regions as query examples. Alan Fekete, Xuemin Lin, redaktorzy, *Australasian Database Conference, ACM*, wolumen 75 serii *CRPIT*, strony 39–47. Australian Computer Society, 2008. (Cytowano na stronach: 105)
- [30] B. Jabłoński. *Filtracja obrazów i trajektorii przestrzennych za pomocą równań różniczkowych cząstkowych*. EXIT, 2008. (Cytowano na stronach: 28)
- [31] Ch. E. Jacobs, A. Finkelstein, D. H. Salesin. Fast multiresolution image querying. strony 277–286, 1995. (Cytowano na stronach: 105)



- [32] D. Karpisz. *Komputerowa analiza obrazu RTG zatok czołowych jako podstawa identyfikacji osób*. Rozprawa doktorska, Politechnika Krakowska, Kraków, 2008. (Cytowano na stronach: 4)
- [33] R. Knowlton. Clinical applications of image registration. I. N. Bankman, redaktor, *Handbook of Medical Imaging: Processing and Analysis*, rozdział 38, strony 613–622. Academic Press, San Diego, 2000. (Cytowano na stronach: 68)
- [34] Z. Latała. *Zastosowanie komputerowej analizy obrazu ultrasonograficznego do badania serca*. Rozprawa doktorska, Politechnika Krakowska, Kraków, 2002. (Cytowano na stronach: 4, 54 i 55)
- [35] Y. Liu, D. Zhang, G. Lu, W. Ma. Region-based image retrieval with perceptual colors. Kiyoharu Aizawa, Yuichi Nakamura, Shin'ichi Satoh, redaktorzy, *Pacific-Rim Conference on Multimedia*, wolumen 3332 serii *Lecture Notes in Computer Science*, strony 931–938. Springer, 2004. (Cytowano na stronach: 45 i 105)
- [36] Y. Liu, D. Zhang, G. Lu, W. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, January 2007. (Cytowano na stronach: 105)
- [37] M. Lorenzo-Valdés, G. Sanchez-Ortiz, R. Mohiaddin, D. Rueckert. Atlas-Based Segmentation and Tracking of 3D Cardiac MR Images Using Non-rigid Registration. wolumen 2488, strony 642–650. Springer, 2002. (Cytowano na stronach: 18 i 68)
- [38] B. D. Lucas, T. Kanade. An iterative image registration technique with an application to stereo vision. Patrick J. Hayes, redaktor, *Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI 1981*, strony 674–679, Vancouver, Canada, 1981. William Kaufmann. (Cytowano na stronach: 122 i 123)
- [39] J. B. Maintz, M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, March 1998. (Cytowano na stronach: 124)
- [40] M. Maire, P. Arbelaez, Ch. Fowlkes, J. Malik. Using contours to detect and localize junctions in natural images. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, strony 1–8. IEEE Computer Society, 2008. (Cytowano na stronach: 117)
- [41] T. Malisiewicz, A. A. Efros. Recognition by association via learning per-exemplar distances. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 0:1–8, 2008. (Cytowano na stronach: 18)



- [42] R. Malladi, J. A. Sethian, B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, February 1995. (Cytowano na stronach: [16](#) i [17](#))
- [43] B. S. Manjunath, W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996. (Cytowano na stronach: [105](#))
- [44] 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. (Cytowano na stronach: [9](#) i [11](#))
- [45] R. Mehrotra, J. E. Gary. Similar-shape retrieval in shape data management. *IEEE Computer*, 28(9):57–62, 1995. (Cytowano na stronach: [105](#))
- [46] H. Müller, N. Michoux, D. Bandon, A. Geissbuhler. A review of content-based image retrieval systems in medical applications - clinical benefits and future directions. *International Journal of Medical Informatics*, 73(1):1–23, 2003. (Cytowano na stronach: [105](#))
- [47] M. Nałęcz, L. Chmielewski, J. L. Kulikowski, A. Nowakowski, redaktorzy. *Obrazowanie Biomedyczne*, wolumen 8 serii *Inżynieria biomedyczna*. Exit, 2004. (Cytowano na stronach: [2](#))
- [48] R. Nock, F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458, Listopad 2004. (Cytowano na stronach: [45](#))
- [49] M. R. Ogiela, R. Tadeusiewicz. *Modern Computational Intelligence Methods for the Interpretation of Medical Image*, wolumen 84 serii *Studies in Computational Intelligence*. Springer Berlin / Heidelberg, 2008. (Cytowano na stronach: [4](#), [47](#), [104](#) i [108](#))
- [50] S. Osher, J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, November 1988. (Cytowano na stronach: [16](#) i [17](#))
- [51] N. Otsu. A threshold selection method from grey-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, January 1979. (Cytowano na stronach: [14](#))

- [52] H. Park, P.H. Bland, C.R. Meyer. Construction of an abdominal probabilistic atlas and its application in segmentation. *IEEE Transactions on Medical Imaging*, 22(4):483–492, April 2003. (Cytowano na stronach: 18 i 68)
- [53] P. Perona, J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990. (Cytowano na stronach: 28 i 48)
- [54] E. G. M. Petrakis. Design and evaluation of spatial similarity approaches for image retrieval. *Image Vision Computing*, 20(1):59–76, 2002. (Cytowano na stronach: 105)
- [55] R. Petryniak. Application of watershed algorithm to image hierarchical segmentation. *Czasopismo Techniczne Politechniki Krakowskiej*, 7:417–424, 2010. (Cytowano na stronach: 97)
- [56] J. Reinhardt, W.E. Higgins. Automatic generation of image-segmentation processes. *IEEE International Conference on Image Processing, 1994. Proceedings. ICIP-94.*, wolumen 3, strony 791–795, Listopad 1994. (Cytowano na stronach: 18)
- [57] X. Ren. Multi-scale improves boundary detection in natural images. *European Conference on Computer Vision*, wolumen 3, strony 533–545, 2008. (Cytowano na stronach: 117)
- [58] T. Rohlfing, Jr. Calvin R. Maurer. Multi-classifier framework for atlas-based image segmentation. *Pattern Recognition Letters*, 26:2070–2079, October 2005. (Cytowano na stronach: 18 i 68)
- [59] Torsten Rohlfing, Robert Brandt, Randolph Menzel, Daniel B. Russakoff, Jr. Calvin R. Maurer. Quo vadis, atlas-based segmentation? Jasjit Suri, David L. Wilson, Swamy Laxminarayan, redaktorzy, *The Handbook of Medical Image Analysis – Volume III: Registration Models*, rozdział 11, strony 435–486. Kluwer Academic / Plenum Publishers, New York, NY, Aug. 2005. (Cytowano na stronach: 18 i 68)
- [60] John C. Russ. *The Image Processing Handbook*. Taylor and Francis, Inc., London, UK, wydanie 5, Wrzesień 2006. (Cytowano na stronach: 23)
- [61] T.G. Ryall, J. Sandor. Statistical pattern matching. *Pattern Recognition Letters*, 9:163–168, 1989. (Cytowano na stronach: 18)
- [62] L. Shafarenko, M. Petrou, J. Kittler. Automatic watershed segmentation of randomly textured color images. *Image Processing, IEEE Transactions on*, 6(11):1530–1544, November 1997. (Cytowano na stronach: 17)

- [63] J. Shi, J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. (Cytowano na stronach: 17)
- [64] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 22(12):1349–1380, 2000. (Cytowano na stronach: 105)
- [65] M. Smiatacz, W. Malina. Active shape models in practice. *Computer Recognition Systems, Advances in Soft Computing*, 30:451–459, 2005. (Cytowano na stronach: 19)
- [66] M. Smiatacz, W. Malina. Aktywne modele kształtu i ich biometryczne zastosowania. *Techniki Komputerowe, Biuletyn Informacyjny*, wolumen 40, strony 43–52. Instytut Maszyn Matematycznych, 2005. (Cytowano na stronach: 19)
- [67] C. O. S. Sorzano, P. Thevenaz, M. Unser. Elastic registration of biological images using vector-spline regularization. *IEEE Transactions on Biomedical Engineering*, 52(4):652–663, 2005. (Cytowano na stronach: 125)
- [68] Z. Tabor. Metody teorii grafów w analizie obrazów. *Wiadomości Stereologiczne, Biuletyn Polskiego Towarzystwa Stereologicznego*, strony 24–32, December 2006. (Cytowano na stronach: 36)
- [69] R. Tadeusiewicz. *Sieci neuronowe. Problemy Współczesnej Nauki i Techniki. Informatyka*. Akademicka Oficyna Wydawnicza RM, 1993. (Cytowano na stronach: 3)
- [70] R. Tadeusiewicz, M.R. Ogiela. *Medical image understanding technology: artificial intelligence and soft-computing for image understanding*. Studies in fuzziness and soft computing. Springer, 2004. (Cytowano na stronach: 4, 47, 104 i 108)
- [71] A. Tremeau, P. Colantoni. Regions adjacency graph applied to color image segmentation. *Image Processing, IEEE Transactions on*, 9(4):735–744, April 2000. (Cytowano na stronach: 17)
- [72] B. C. Vemuri, J. Ye, Y. Chen, C. M. Leonard. Image registration via level-set motion: applications to atlas-based segmentation. *Medical Image Analysis*, 7(1):1–20, March 2003. (Cytowano na stronach: 125)
- [73] L. Vincent, P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991. (Cytowano na stronach: 16, 17, 34 i 35)

- 
- [74] Paul Viola, Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, Maj 2004. (Cytowano na stronach: [105](#))
- [75] L. Wojnar, K. J. Kurzydłowski, J. Szala. *Praktyka analizy obrazu*. Polskie Towarzystwo Stereologiczne, Kraków, 2002. (Cytowano na stronach: [12](#) i [49](#))
- [76] Y. Xu, E. C. Uberbacher. 2d image segmentation using minimum spanning trees. *Image Vision Computing*, 15(1):47–57, January 1997. (Cytowano na stronach: [36](#))
- [77] J-Ch Yen, F-J Chang, S. Chang. A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing*, 4(3):370–378, 1995. (Cytowano na stronach: [14](#))
- [78] Th. Zahariadis, D. Kalivas. A spiral search algorithm for fast estimation of block motion vectors. *Proceedings of the VIII European Signal Processing Conference (EUSIPCO-96)*, wolumen 2, strony 1079–1082, Trieste, Italy, 1996. (Cytowano na stronach: [122](#) i [123](#))
- [79] B. Zitová, J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003. (Cytowano na stronach: [124](#))

---

## *Źródła internetowe*

- [80] bUnwarpJ - ImageJ plugin for bidirectional elastic registration of two images. [Oprogramowanie]. <http://biocomp.cnb.uam.es/~iarganda/bUnwarpJ/>.
- [81] GTK+ - Biblioteka C programowania interfejsów użytkownika. [Oprogramowanie]. <http://www.gtk.org/>.
- [82] ImageJ - Image Processing and Analysis in Java. [Oprogramowanie]. <http://rsbweb.nih.gov/ij/>.
- [83] imgSeek - Content-based photo collection manager. [Oprogramowanie]. Ricardo Niederberger Cabral. <http://www.imgseek.net/>.
- [84] Insight Segmentation and Registration Toolkit (ITK). [Oprogramowanie]. <http://itk.org/>.
- [85] Itk levelset motion registration filter. [http://www.itk.org/Doxygen/html/classitk\\_1\\_1LevelSetMotionRegistrationFilter.html](http://www.itk.org/Doxygen/html/classitk_1_1LevelSetMotionRegistrationFilter.html).
- [86] Qt - A cross-platform application and UI framework. [Oprogramowanie]. Nokia Corporation. <http://qt.nokia.com/>.
- [87] Rafał Petryniak, Oficjalna strona internetowa. <http://sites.google.com/site/rafalpetryniakresearchpl/>.
- [88] SQLite - Software library that implements SQL database engine. [Oprogramowanie]. <http://www.sqlite.org/>.
- [89] The CImg Library - C++ Template Image Processing Toolkit. [Oprogramowanie]. Greyc Laboratory. <http://cimg.sourceforge.net/>.
- [90] Pandore: A library of image processing operators (Version 6.4). [Oprogramowanie]. Greyc Laboratory. <http://www.greyc.ensicaen.fr/EquipeImage/Pandore>, 2010. [accessed April 2010].
- [91] Wikipedia. Dicom – Wikipedia, the free encyclopedia. <http://pl.wikipedia.org/w/index.php?title=DICOM&oldid=22049809>, 2010. [Materiał w sieci; wersja z dnia 14.06.2010].

---

## Skorowidz

- łączenie regionów, [16](#), [17](#), [39](#)
  - popularne kryteria, [41](#)
  - przykłady użycia, [43](#)
  - reprezentacja grafowa, [39](#)
- aktywne kontury, [16](#)
- automatyczne rozumienie obrazów, [4](#), [47](#),  
[104](#), [108](#)
- binaryzacja obrazu
  - progowanie adaptacyjne, [16](#)
  - progowanie z histerezą, [14](#), [118](#)
- detekcja krawędzi
  - algorytm Canny, [56](#), [57](#), [59](#), [60](#), [62–67](#),  
[82](#), [117](#), [119](#)
  - kryteria działania, [117](#)
  - zasady działania, [117](#)
- diagnostyka obrazowa, [2](#), [3](#), [18](#), [103](#)
- duże obiekty, [4–6](#), [8](#), [14](#), [16](#), [21](#), [33](#), [38](#), [39](#),  
[43](#), [45](#), [47](#), [50](#), [56](#), [57](#), [64](#), [66–68](#),  
[92–94](#), [102](#), [103](#), [107–112](#)
- kryteria, [5](#), [6](#)
- modele, [19](#), [35](#), [47](#), [68](#), [110](#), [112](#)
- problemy, [6](#), [8](#), [14](#), [112](#)
- przykłady, [6](#), [8](#), [11](#), [64](#)
- falki Haara, [105](#), [107](#)
- generowanie algorytmów, [18](#)
- gradient promieniowy, [54](#)
- indeksacja, [40](#), [74](#), [107](#)
- metody automatyczne, [3](#), [4](#), [7](#), [18](#), [19](#), [92](#),  
[93](#), [103](#), [104](#), [106](#), [109](#), [111–113](#),  
[124](#)
- metody manualne, [6](#), [35](#), [54](#), [68](#), [69](#), [93](#),  
[104](#), [105](#), [112](#), [124](#)
- nakładanie obrazów, [68](#), [69](#), [71–74](#), [78](#), [82](#),  
[84](#), [92](#), [102](#), [120](#), [124–128](#)
- obrazy
  - baza z Aachen, [11](#)
  - baza z Berkeley, [11](#)
  - fotografie natury, [8](#), [64](#), [93](#), [100](#)
  - radiogramy kości piszczelowej, [84](#), [88](#)
  - radiogramy kości udowej, [8](#), [11](#),  
[13–15](#), [17](#), [37](#), [38](#), [43–45](#), [52](#), [57](#),  
[64](#), [80–84](#), [89](#), [97](#), [102](#), [106](#)
  - radiogramy lewej dłoni, [84](#), [87](#)
  - radiogramy miednicy, [84](#), [86](#)
  - radiogramy ręki, [84](#), [85](#)
  - radiogramy ramienia, [84](#), [91](#)
  - radiogramy stopy, [84](#), [90](#)
  - skany tomograficzne płuc, [8](#), [9](#), [11](#),  
[27](#), [43](#), [44](#), [66](#), [68](#), [69](#), [78](#), [80](#), [102](#),  
[105–107](#), [124](#), [126](#)
- odwołania do stron internetowych
  - Rafał Petryniak - strona domowa, [43](#),  
[103](#), [107](#)
  - Wikipedia: DICOM, [104](#)
- pocienianie, [50](#)



- pomiary obiektów, [4](#), [12](#), [18](#), [22](#), [41](#), [71](#),  
[105](#)
- promieniowanie rentgenowskie, [2](#)
- przepływ optyczny, [122](#)
  - algorytm Horn-Schunck, [122](#), [123](#)
  - algorytm korelacyjny dopasowania  
bloków, [122](#), [123](#)
  - algorytm Lucas-Kanade, [122](#), [123](#)
- przygotowane oprogramowanie
  - graficzna bazy danych, [103–105](#)
  - funkcje systemu, [108](#)
  - koncepcja, [103](#)
  - realizacja, [107](#)
  - interaktywne łączenie segmentów, [43](#)
- rozprawa
  - cel, [8](#), [19](#)
  - podsumowanie, [110](#)
  - teza, [19](#)
  - układ treści, [10](#)
  - wnioski, [112](#)
- segmentacja obrazu
  - aktywne modele kształtu, [18](#)
  - algorytm działów wodnych, [16](#), [34](#), [93](#)
    - przykłady użycia, [99](#)
    - segmentacja hierarchiczna, [97](#)
    - znaczniki ciągłe, [94](#), [96](#), [97](#)
    - znaczniki punktowe, [94](#), [95](#)
  - algorytmy grafowe, [36](#)
  - algorytmy rozrostu, [16](#)
  - bazująca na atlasie, [18](#)
- szkieletyzacja, [50](#)
- użyte narzędzia
  - bUnwarpJ, [70](#), [125](#), [127](#), [128](#)
  - CImg, [107](#)
  - GTK+, [107](#)
  - ImageJ, [125](#)
  - imgSeek, [105](#)
  - Insight Toolkit, [125](#), [126](#)
    - LevelSetMotionRegistrationFilter,  
[125](#)
  - Pandore, [43](#)
  - Qt, [107](#)
  - SQLite, [107](#)
- usuwanie szumu
  - model Perona-Malik, [28](#)
  - regularyzacja modelu nieliniowego,  
[30](#)
  - rozmycie Gaussa, [15](#), [24](#), [25](#), [27](#), [29](#),  
[30](#), [34](#), [35](#), [38](#), [43](#), [52](#), [57](#), [65](#), [74](#),  
[77](#), [78](#), [80](#), [93](#), [117](#), [118](#)
  - funkcja Gaussa, [26](#)
  - macierz filtru 7x7, [27](#)
  - wykres funkcji, [26](#)

Wersja elektroniczna rozprawy jak również dodatkowe materiały do niej dołączone zostały zamieszczone na stronie internetowej autora pod następującym adresem:

[http://sites.google.com/site/rafalpetryniakresearchpl/  
list-of-publications/phd\\_thesis](http://sites.google.com/site/rafalpetryniakresearchpl/list-of-publications/phd_thesis)