



Akademia Górniczo-Hutnicza  
im. Stanisława Staszica  
w Krakowie



Wydział Elektrotechniki, Automatyki,  
Informatyki i Inżynierii Biomedycznej

Katedra Metrologii i Elektroniki

Rozprawa doktorska

**Komputerowy model symulacyjny zabiegu  
bronchoskopii oraz jego zastosowanie  
do opracowania i testowania metod estymacji  
przesunięcia końcówki bronchofiberoskopu**

mgr inż. Mirosław Socha

Promotor: prof. dr hab. inż. Tomasz P. Zieliński

Kraków 2013

*Chciałbym w tym miejscu podziękować:*

*prof. dr hab. inż. Tomaszowi P. Zielińskiemu  
za podjęcie obowiązków promotora  
oraz za cenne rady i wskazówki,  
dzięki którym ta praca powstała,*

*dr med. Mariuszowi Dupładze  
za inspiracje i wieloletnią współpracę,*

*pracownikom Katedry Metrologii i Elektroniki  
za życzliwość, która pomogła mi w realizacji tej pracy,  
a szczególnie prof. dr hab. inż. Januszowi Gajdzie  
za mobilizowanie do rozwoju i okazaną cierpliwość,  
dr inż. Waławowi Gawędzkiemu,  
dr inż. Andrzejowi Skalskiemu,  
dr hab. inż. Krzysztofowi Dudzie,  
dr inż. Jarosławowi Bułatowi,  
dr inż. Pawłowi Turczy,  
dr inż. Dariuszowi Borkowskiemu  
za wszelką okazaną pomoc i wsparcie*

*oraz Rodzinie  
za to, że zawsze jest przy mnie.*



# SPIS TREŚCI

## Spis treści

i

<b>1</b>	<b>Wstęp</b>	<b>1</b>
1.1	Motywacja . . . . .	1
1.2	Uzasadnienie ważności i celowości podejmowanej tematyki . . . . .	2
1.3	Przewidywane rezultaty rozprawy doktorskiej . . . . .	2
1.4	Struktura pracy . . . . .	2
1.5	Projekty badawcze i publikacje autora . . . . .	3
<b>2</b>	<b>Aktualny stan wiedzy medycznej i technicznej</b>	<b>6</b>
2.1	Układ oddechowy człowieka . . . . .	6
2.1.1	Budowa tchawicy i oskrzeli . . . . .	7
2.1.2	Węzły chłonne . . . . .	8
2.1.3	Rozmieszczenie węzłów chłonnych w klatce piersiowej . . . . .	9
2.2	Bronchoskopia . . . . .	10
2.3	Tomografia komputerowa . . . . .	15
2.3.1	Zasada działania tomografii komputerowej . . . . .	16
2.3.2	Interpretacja danych . . . . .	17
2.4	Wspomaganie zabiegów . . . . .	19
<b>3</b>	<b>Model zabiegu bronchofiberoskopii</b>	<b>26</b>
3.1	Wirtualna bronchoskopia . . . . .	26
3.2	Model wirtualnego fantomu drzewa oskrzelowego . . . . .	28
3.2.1	Modelowanie kształtu . . . . .	29
3.2.1.1	Szkielet drzewa oskrzelowego . . . . .	31
3.2.1.2	Dodanie szczegółów . . . . .	31
3.2.1.3	Pogrubienie szkieletu . . . . .	34
3.2.2	Rekonstrukcja powierzchni . . . . .	35
3.2.3	Modelowanie wyglądu powierzchni . . . . .	36
3.2.3.1	Model cieniowania . . . . .	37
3.2.3.2	Mapowanie tekstur . . . . .	46
3.3	Model kamery endoskopu . . . . .	49
3.3.1	Dystorsja obrazu . . . . .	51
3.3.2	Identyfikacja parametrów modelu zniekształceń . . . . .	51
3.3.3	Implementacja wirtualnego modelu kamery . . . . .	55
3.4	Model oświetlenia endoskopu . . . . .	58
3.5	Podsumowanie . . . . .	60

<b>4</b>	<b>Metoda śledzenia końcówki bronchofiberoskopu</b>	<b>61</b>
4.1	Przegląd metod . . . . .	61
4.2	Opracowana metoda śledzenia bronchoskopu . . . . .	65
4.2.1	Analiza ruchu kamery . . . . .	67
4.3	Szacowanie przesunięcia endoskopu . . . . .	68
4.3.1	Wyznaczanie ruchu endoskopu . . . . .	70
4.3.2	Koszty obliczeniowe algorytmu . . . . .	73
4.3.3	Korelacja . . . . .	74
4.4	Dokładne dopasowywanie obrazów metodą informacji wzajemnej . . . . .	75
4.5	Weryfikacja metody . . . . .	76
4.6	Akceleracja obliczeń w technologii CUDA . . . . .	80
4.6.1	Przygotowanie danych . . . . .	80
4.6.2	Korelacja obrazów . . . . .	82
4.6.3	Analiza wydajności . . . . .	84
<b>5</b>	<b>Podsumowanie</b>	<b>87</b>
5.1	Uzyskane wyniki końcowe . . . . .	87
5.2	Oryginalne osiągnięcia . . . . .	90
<b>A</b>	<b>Zagadnienia informatyczne</b>	<b>91</b>
A.1	Biblioteka VTK . . . . .	91
A.1.1	Architektura . . . . .	92
A.1.2	Strumień wizualizacji . . . . .	92
A.1.3	Model graficzny . . . . .	94
A.1.4	Podstawowe obiekty modelu graficznego VTK . . . . .	95
A.1.5	Reprezentacja danych . . . . .	96
A.1.6	Typy atrybutów danych . . . . .	99
A.1.7	Typy zbiorów danych . . . . .	101
A.1.8	Podsumowanie . . . . .	102
A.2	Biblioteka CUDA . . . . .	103
A.2.1	Przetwarzanie danych w GPU . . . . .	105
A.2.2	Organizacja pamięci . . . . .	108
A.2.3	Interfejs programisty . . . . .	110
A.2.4	Podsumowanie . . . . .	111
	<b>Bibliografia</b>	<b>112</b>
	<b>Spis rysunków</b>	<b>120</b>
	<b>Spis tablic</b>	<b>124</b>

## WSTĘP

### 1.1 Motywacja

Bronchofiberoskopia jest małoinwazyjnym badaniem pozwalającym wzrokowo ocenić przy pomocy giętkiego endoskopu – bronchofiberoskopu – stan górnych dróg oddechowych człowieka. Jednym z głównych zabiegów współczesnej diagnostyki zmian chorobowych klatki piersiowej, wykonywanych podczas bronchoskopii, jest przezoskrzelowa biopsja aspiracyjna. Zabieg ten polega na pobraniu za pomocą cienkiej igły, poprzez aspirację, tkanek z powiększonych węzłów chłonnych znajdujących się za ścianami drzewa oskrzelowego. Próbki takie są następnie poddawane analizie histopatologicznej w celu stwierdzenia obecności tkanek nowotworowych. Obecnie, lekarz endoskopista wybiera dogodnie miejsce wykonania biopsji jedynie na podstawie: klisz z przekrojami klatki piersiowej pacjenta (obrazy uzyskane w tomografii komputerowej TK), własnej wyobraźni przestrzennej oraz doświadczenia. Z tych powodów pobranie tkanek z węzłów chłonnych jest zadaniem trudnym, szczególnie w tchawicy oraz w przypadku niewielkich rozmiarów węzłów.

Komputerowy system planowania i wspomaganie zabiegów bronchoskopowych ma umożliwić pełniejsze wykorzystanie danych z wykonywanego przed bronchofiberoskopią badania tomografem komputerowym. Dzięki interaktywnej wizualizacji dwuwymiarowych przekroi przez ciało pacjenta oraz trójwymiarowym reprezentacjom organów, lekarz będzie mógł dokładniej przeanalizować dostępne dane przed zabiegiem, np. odszukać powiększone węzły chłonne oraz zaplanować zabieg bronchofiberoskopii.

Algorytmy estymacji położenia końcówki bronchofiberoskopu na podstawie obrazów z endoskopu, mają kluczowe znaczenie w komputerowych systemach wspomaganie zabiegów bronchofiberoskopii. Poprzez analizę sekwencji obrazów rzeczywistych oraz wirtualnych, systemy te umożliwią określenie położenia końcówki bronchofiberoskopu w trakcie zabiegu. Dzięki temu, możliwe będzie precyzyjne ustawienie końcówki endoskopu w bezpośrednim sąsiedztwie węzła chłonnego, a to z kolei umożliwi poprawę skuteczności zabiegów biopsji przezoskrzelowej. Co ważne, poprawa skuteczności zabiegów możliwa będzie bez inwestowania w nowy, specjalistyczny sprzęt medyczny, ponieważ algorytmy będą bazować na danych, które zazwyczaj są dostępne podczas zabiegów biopsji: obrazie wideo z endoskopu oraz danych z tomografii komputerowej.

## 1.2 Uzasadnienie ważności i celowości podejmowanej tematyki

Wczesna diagnostyka zmian nowotworowych ma kluczowe znaczenie dla przeżywalności pacjentów. Dlatego niezbędne jest rozwijanie wszelkich metod wspomaganie takiej diagnostyki. Estymacja położenia końcówki bronchofiberoskopu jest zagadnieniem rozwijanym przez firmy zajmujące się produkcją aparatury do wykonywania zabiegów bronchofiberoskopii. Są to rozwiązania sprzętowe, które bazują głównie na dodatkowych czujnikach umieszczanych w końcówce endoskopu. Rozwiązania te są kosztowne i wymagają zakupu dodatkowego sprzętu.

Proponowana w niniejszej pracy metoda wspomaganie zabiegu biopsji nie wymaga użycia żadnego dodatkowego sprzętu, bazuje jedynie na rejestrowanych przez kamerę obrazach z wideobronchofiberoskopu oraz na przestrzennych danych pacjenta, pochodzących z wcześniej wykonanej tomografii komputerowej. Brak dodatkowego sprzętu medycznego jest niewątpliwą zaletą proponowanego rozwiązania.

## 1.3 Przewidywane rezultaty rozprawy doktorskiej

Praca ma na celu opracowanie symulacyjnego modelu komputerowego systemu nawigacyjnego, umożliwiającego wspomaganie zabiegów bronchofiberoskopii poprzez estymowanie położenia końcówki endoskopu. Model systemu będzie obejmował przestrzenny model górnych dróg oddechowych oraz algorytmy estymacji położenia końcówki endoskopu. Określenie położenia końcówki endoskopu w drzewie oskrzelowym ma kluczowe znaczenie w systemach wspomaganie zabiegów medycznych, szczególnie podczas transbronchialnej biopsji aspiracyjnej wykonywanej przy diagnostyce nowotworów klatki piersiowej.

Opracowany zostanie specjalizowany *algorytm estymacji położenia końcówki endoskopu*, zoptymalizowany do specyficznej budowy tchawicy i oskrzeli (brak krawędzi), dostosowany do źródła danych (szerokokątnej, nieliniowej optyki kamery endoskopu) oraz specyficznych zmienności kształtu (oddychanie, bicie serca, odkrztuszanie). Algorytm ten umożliwi estymowanie położenia końcówki bronchofiberoskopu w wirtualnym modelu na podstawie sekwencji obrazów rzeczywistych oraz danych z tomografii komputerowej pacjenta.

Przewiduje się również *opracowanie przestrzennego, w pełni konfigurowalnego, wirtualnego modelu górnych dróg oddechowych człowieka*, który umożliwi generowanie obrazów wirtualnej bronchoskopii zbliżonych do obrazów rzeczywistych. Model ten umożliwi testowanie oraz weryfikację algorytmów estymacji położenia endoskopu. Możliwe również będzie wykorzystanie modelu do celów szkoleniowych (tzw. wirtualny fantom).

## 1.4 Struktura pracy

Praca została podzielona na sześć części. Pierwszy rozdział zawiera informacje wprowadzające do tematyki rozprawy: motywację, cel i przewidywane rezultaty pracy oraz krótkie omówienie dorobku naukowego autora. Następnie, w rozdziale drugim, przedstawiono zagadnienia medyczne związane z tematyką rozprawy. Rozdział trzeci zawiera opis opracowanego przez autora komputerowego modelu zabiegu bronchofiberoskopowego. Natomiast rozdział czwarty poświęcony jest metodzie śledzenia końcówki endoskopu, bazującej na porównywaniu rzeczywistych obrazów z endoskopu z obrazami wirtualnymi generowanymi na podstawie danych

z tomografii komputerowej. Rozdział piąty zawiera podsumowanie pracy. W dodatku zawarto opis bibliotek wykorzystanych w pracy: biblioteki Visualization Toolkit do prezentacji danych oraz biblioteki nVidia CUDA (ang. *Compute Unified Device Architecture*) użytej do akceleracji obliczeń w estymacji położenia końcówki endoskopu. Na końcu rozprawy znajduje się wykaz tekstów źródłowych, spis rysunków oraz tabel.

## 1.5 Projekty badawcze i publikacje doktoranta z zakresu tematyki rozprawy

Autor rozprawy wspólnie z lekarzem pulmonologiem, dr. med. Mariuszem Duplągą oraz współpracownikami z Katedry Metrologii i Elektroniki AGH oraz Katedry Telekomunikacji AGH od roku 2004 rozwija systemy wspomaganie zabiegów bronchoskopii, które bazowałyby jedynie na dostępnych danych: obrazie wideo rejestrowanym podczas zabiegu oraz danych z tomografii komputerowej wykonywanej przed zabiegiem bronchoskopii. Opracowane systemy wspomaganie zabiegów bronchoskopii nie wymagają zastosowania dodatkowego sprzętu używanego bezpośrednio na sali zabiegowej, który wymagałby zatwierdzenia przez komisje lekarskie odpowiedzialne za bezpieczeństwo pacjenta. Prace skupiały się na maksymalnym wykorzystaniu zasobów standardowej pracowni bronchoskopowej.

W wyniku prac zespołu przebadano metody estymacji ruchu końcówki bronchoskopu bazujące na obrazie wideo. Pierwsze badania w tej tematyce zrealizowano w projekcie KBN nr 3 T11E 039 27 pod tytułem „*Opracowanie modelu bronchoskopowego systemu nawigacyjnego opartego na analizie obrazów endoskopowych i wirtualnych*”. Projekt ten zrealizowany został w latach 2004-2006 w Katedrze Metrologii Akademii Górniczo-Hutniczej, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki. Autor rozprawy brał w nim udział jako wykonawca i zajmował się głównie zagadnieniami generowania obrazów wirtualnej bronchoskopii oraz identyfikacją i modelowaniem zniekształceń optyki endoskopu. Opracowany algorytm estymacji położenia endoskopu został opisany między innymi w pracach (Twardowski i inni, 2006; Duda i inni, 2006; Bułat i inni, 2006, 2008). Ze względu na niskie koszty obliczeniowe, algorytm ten miał szansę pracować w czasie rzeczywistym. Głównym zadaniem autora niniejszej pracy było opracowanie metod szybkiego generowania wirtualnych obrazów drzewa oskrzelowego oraz przyspieszenie kluczowych fragmentów algorytmu.

Opracowana w 2006 roku szybka metoda estymacji położenia końcówki endoskopu, opisana w pracy (Twardowski i inni, 2006) nie była w stanie pracować w czasie rzeczywistym — czas przetwarzania danych dla jednej klatki filmu z endoskopu wynosił ponad 620ms, podczas gdy klatki napływały co 40ms (25 ramek na sekundę). W pracy (Bułat i inni, 2010) przedstawiono możliwe sposoby skrócenia czasu obliczeń. Wskazywano na możliwość zastosowania przetwarzania równoległego na wielu procesorach CPU poprzez mechanizmy wymiany komunikatów MPI (ang. *Message Passing Interface*). Metoda komunikacji między różnymi częściami medycznego systemu informatycznego została z powodzeniem przetestowana i sprawdzona podczas tworzenia rozbudowanego systemu *Bronchovid* (Dupląga i inni, 2010). Przyspieszenie obliczeń z wykorzystaniem MPI wiąże się jednak z użyciem kosztownych i energochłonnych komputerów wieloprocessorowych, zaś uzyskiwane przyspieszenie nie jest zadowalające. Inną metodą jest zoptymalizowanie kodu pod kątem wykorzystania rozkazów SSE (ang. *Streaming SIMD Extensions*) procesora CPU, które pozwalają znacznie szybciej wykonywać obliczenia matematyczne. Współczesne kompilatory używają rozszerzonej listy rozkazów multimedialnych, pozwalając w pełni wykorzystać możliwości procesorów. Również wykorzystanie

w części algorytmu dedykowanej biblioteki numerycznej BLAS (ang. *Basic Linear Algebra Subprograms*) umożliwiłoby zwiększenie wydajności. Wśród zaproponowanych metod akceleracji znalazła się również możliwość użycia wielordzeniowych procesorów graficznych GPU. W dalsze części rozprawy zostanie omówiona szybka metoda estymacji położenia endoskopu oraz zastosowane metody jej akceleracji z użyciem biblioteki CUDA umożliwiającej obliczenia na GPU.

W badaniach przeprowadzonych przez autora rozprawy, biblioteka Visualization Toolkit (VTK) posłużyła głównie do implementacji wydajnego i elastycznego modelu drzewa oskrzelowego. W implementacji wykorzystano strumieniowe możliwości przetwarzania danych — zbudowano strumień umożliwiający wygenerowanie powierzchni wirtualnego drzewa oskrzelowego, a następnie jego wydajną wizualizację. Wykorzystano wbudowane typy zbiorów danych, głównie *poly-data*, oraz możliwości sterowania procesem tworzenia obrazu poprzez użycie obiektów modelu graficznego VTK.

Kolejnym projektem, w którym autor brał udział był projekt rozwojowy MNiSzW nr R13 011 03 pod tytułem „*BRONCHOVID - Zintegrowany system wspomagający diagnostykę badań bronchoskopowych przez cyfrową archiwizację, semantyczne porządkowanie zapisów wizyjnych, wykrywanie i wyszukiwanie treści istotnych diagnostycznie oraz interaktywną wizualizację z użyciem danych tomografii komputerowej*” (Duplaga i inni, 2010). Realizowany był on w Katedrze Telekomunikacji AGH wspólnie z pracownikami Katedry Automatyki AGH, Katedry Metrologii AGH, Instytutu Radioelektroniki Politechniki Warszawskiej oraz Wydziału Lekarskiego, Katedry Pulmonologii, Collegium Medicum UJ. Opracowany system integruje również dane z tomografii komputerowej pacjenta i umożliwia wizualizację oraz analizę danych w celu zaplanowania zabiegu biopsji. Zadaniem autora rozprawy było opracowanie modułu do przestrzennej wizualizacji danych z tomografii komputerowej, integracja i opracowanie interfejsów użytkownika aplikacji do przeglądania i opisywania zapisów wideo oraz opracowanie algorytmów do segmentacji danych (Skalski i inni, 2010).

W latach 2006-2010 autor rozprawy uczestniczył również w pracach nad opracowaniem aktywnej kapsułki endoskopowej, realizowanej w ramach projektu Komisji Europejskiej No 3E061105 „*VECTOR: Versatile Endoscopic Capsule for gastrointestinal TumOr Recognition and therapy*”. Wśród głównych zadań zrealizowanych można wyróżnić opracowanie algorytmów do czyszczenia, segmentacji i wizualizacji układu pokarmowego (Bułat i inni, 2007; Skalski i inni, 2007a,b, 2011).

W latach 2010-2012 doktorant prowadził badania nad:

- komputerową wizualizacją drzewa oskrzelowego na podstawie danych z tomografii komputerowej i akceleracją obliczeń, z tym związanych,
- stworzeniem parametrycznego modelu drzewa oskrzelowego, umożliwiającego generowanie możliwie jak najwierniejszej sekwencji obrazów tzw. wirtualnej bronchoskopii, przypominającej rzeczywistą bronchoskopię,
- szybkim estymowaniem położenia końcówki endoskopu na podstawie metody wcześniej opracowanej zespołowo, która polegała na synchronizacji obrazów rzeczywistych (z rzeczywistych zabiegów bronchoskopii) z obrazami syntetycznymi, generowanymi na podstawie danych z tomografii komputerowej.

Autor rozprawy wykorzystując zdobyte doświadczenie w dziedzinie tworzenia komputerowych systemów do przetwarzania i wizualizacji danych medycznych napisał skrypt pod tytułem *Wizualizacja danych medycznych* (Socha, 2011b). Pozycja ta kierowana jest do studentów studiów II stopnia, pragnących poszerzyć swoją wiedzę na temat prezentacji danych.

W roku 2012 doktorant był jednym z zaproszonych prelegentów, którzy zademonstrowali pierwsze w Polsce interaktywne wizualizacje wirtualnej bronchoskopii prezentowanej w technice 3D (z użyciem pasywnej stereoskopii) na dwóch konferencjach szkoleniowych pt. „II Konferencja Naukowo-Szkoleniowa, Choroby układu oddechowego u dzieci – problemy diagnostyki i terapii”, która odbyła się w dniach 2-3.03.2012 w Zakopanem, oraz „III Konferencja Naukowo-Szkoleniowa, Choroby Układu Oddechowego – problemy diagnostyki i terapii” w Niepołomicach 23-24.03.2012.

## AKTUALNY STAN WIEDZY MEDYCZNEJ I TECHNICZNEJ

W rozdziale tym przedstawione zostaną informacje teoretyczne o charakterze medycznym i technicznym, związane z tematem rozprawy. Omówiona zostanie budowa drzewa oskrzelowego, zabieg bronchoskopii oraz dostępne dane medyczne umożliwiające wspomaganie planowania oraz wykonania zabiegu bronchoskopii.

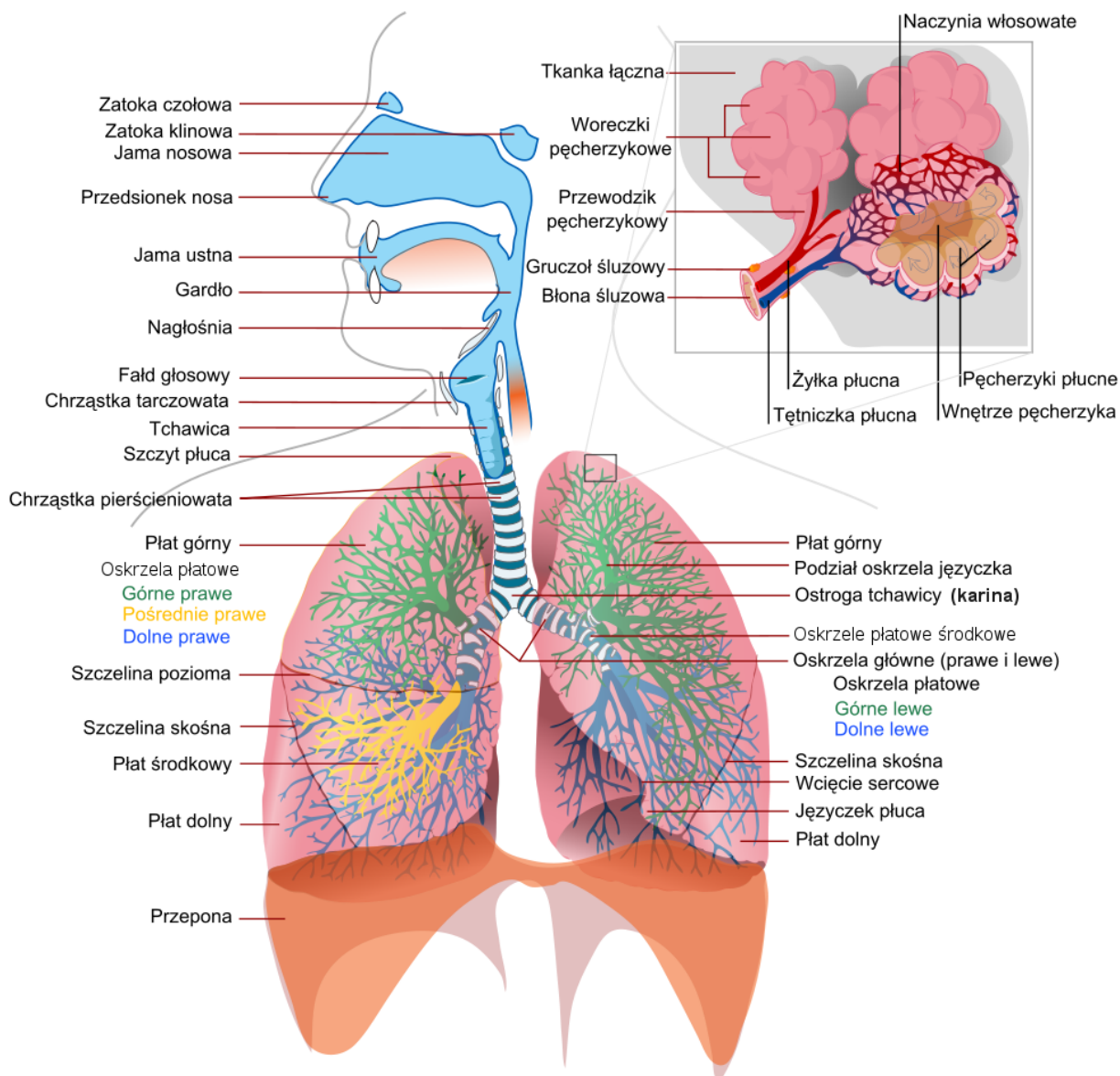
### 2.1 Układ oddechowy człowieka

Układ oddechowy człowieka umożliwia dostarczanie tlenu z powietrza do krwi oraz usuwanie dwutlenku węgla z krwi do powietrza. Tworzą go: *górne drogi oddechowe*, *dolne drogi oddechowe* oraz część zwana *końcowymi drogami oddechowymi*, w której dochodzi do wymiany gazów (Stevens i Lowe, 2000). Na rysunku 2.1 przedstawiono, w sposób schematyczny, budowę układu oddechowego człowieka.

Na górne drogi oddechowe składają się: nozdrza, jama nosowa, nosogardziel, nagłośnia oraz krtań. Dolne drogi oddechowe to: tchawica, oskrzela główne, oskrzela płatowe i segmentowe zakończone oskrzelikami. Głównym zadaniem górnych dróg oddechowych jest filtracja powietrza z drobnych zanieczyszczeń oraz ogrzanie i nawilżenie powietrza. Krtań umożliwia komunikację werbalną poprzez wydawanie dźwięków, zaś umieszczone w jamie nosowej komórki węchowe odczuwanie wrażeń węchowych. Za granicę pomiędzy górnymi oraz dolnymi drogami oddechowymi uważana jest krtań.

Wymiana gazów odbywa się w końcowych obszarach układu oddechowego w zakończeniach oskrzelików zwanych *woreczkami pęcherzykowymi*, w których to powietrze ma bezpośredni kontakt z cienkościennejmi naczyniami krwionośnymi. Powietrze transportowane jest między górną częścią układu oddechowego a jego końcowym obszarem z wykorzystaniem dolnych dróg oddechowych, na które składają się tchawica oraz oskrzela.





Rysunek 2.1: Budowa układu oddechowego człowieka (autor grafiki: Mariana Ruiz Villarreal, wikipedia)

### 2.1.1 Budowa tchawicy i oskrzeli

Tchawica (ang. *trachea*) stanowi kontynuację układu oddechowego i znajduje się za krtanią. Tchawica ma postać dość sztywnej rury, która biegnie ku dołowi i rozdziela się na dwa *oskrzela główne*, zasilające lewe oraz prawe płuco w powietrze (rysunek 2.1). Oskrzela główne dzielą się następnie na oskrzela płatowe, segmentowe oraz oskrzeliki. Tchawica wraz z oskrzelami nazywana jest *drzewem oskrzelowym*.

Zdolność do zachowania kształtu tchawicy wynika z faktu występowania od 15 do 20 niepełnych chrząstnych pierścieni, które obejmują 70-80% obwodu tchawicy. Przerwa między chrząstkami znajduje się od strony kręgosłupa i wypełnia ją *mięsień tchawiczy*, który umożliwia w pewnym zakresie zwięźanie średnicy tchawicy, np. podczas odksztuszania wydzielin. Przekrój poprzeczny tchawicy może mieć kształt zbliżony do okręgu lub też może być niesymetryczny, tzw. „katedralny”. Chrząstki tworzą wówczas sklepienie w kształcie odwróconej litery U

zamkniętej płaską powierzchnią, wyznaczoną przez mięsień tchawicy. Chrząstki występujące w ścianie tchawicy oraz oskrzelach połączone są ze sobą błoną włóknistą i tworzą pofalowaną powierzchnię. Sposób i głębokość pofalowania, kształt i liczba chrząstek są parametrami silnie uwarunkowanymi osobniczo.

Długość tchawicy osoby dorosłej wynosi około 10-12cm, zaś jej średnica waha się w granicach 10-25mm (Stevens i Lowe, 2000). Średnica ta maleje wraz z kolejnymi podziałami na mniejsze oskrzela.

W dolnej części tchawica dzieli się na dwa *oskrzela główne* (duże), które znajdują się w przestrzeni poza płucami i mają w tym miejscu największą średnicę. Oskrzelko główne prawe jest grubsze oraz krótsze od lewego. Dodatkowo oskrzelko prawe przebiega bardziej pionowo, zaś oskrzelko lewe jest ułożone zazwyczaj bardziej poziomo.

W miejscu podziału tchawicy na oskrzela główne, na wysokości czwartego i piątego kręgu piersiowego, znajduje się ostroga tchawicy zwana *kariną* (łac. *carina trachea*). Oskrzelko główne wchodzi do płuc wraz z *tętnicami płucnymi* i dzieli się wówczas na *oskrzela płatowe* zasilające pięć płatów płucnych. Prawe oskrzelko główne dzieli się na trzy oskrzela płatowe: *oskrzelko płata prawego górnego*, *oskrzelko płata środkowego* oraz *oskrzelko płata dolnego*. Lewe oskrzelko zasilają w powietrze dwa płaty poprzez: *oskrzelko płata lewego górnego* oraz *oskrzelko płata lewego dolnego*. Wszystkie oskrzela płatowe dzielą się następnie na różną liczbę *oskrzeli segmentowych*, które dzieląc się jeszcze wielokrotnie, ostatecznie tworzą *oskrzeliki zakończone pęcherzykami płucnymi* (rysunek 2.1) (Stevens i Lowe, 2000).

Zmiana wymiarów układu oddechowego, w postaci np. lokalnego zwężenia tchawicy, kariny, oskrzeli głównych lub innych ostróg, może świadczyć o powiększeniu węzłów chłonnych. A to z kolei sugeruje, że w okolicznych tkankach może występować stan zapalny lub rozwija się proces nowotworowy.

### 2.1.2 Węzły chłonne

Ważnym elementem układu immunologicznego, którego głównym zadaniem jest ochrona organizmu przed chorobami i infekcjami są węzły chłonne (łac. *nodi lymphatici*). Poprzez naczynia limfatyczne do węzłów chłonnych sphywa płyn międzykomórkowy (limfa). Zadaniem węzłów chłonnych jest filtracja limfy z drobnych cząstek, takich jak mikroorganizmy czy też np. pył węglowy wchłonięty przez płuca. Filtracja odbywa się w komórkach fagocytarnych (Stevens i Lowe, 2000). Po odfiltrowaniu limfa wraca do układu limfatycznego, zaś wychwycone cząstki są usuwane przez makrofagi (bez wywoływania odpowiedzi immunologicznej) oraz komórki limfoidalne (poprzez odpowiedź immunologiczną, w której biorą udział antygeny i komórki limfoidalne).

Nieaktywny węzeł chłonny ma niewielkie wymiary (kilka milimetrów), jednak w przypadku wystąpienia dużej liczby „zanieczyszczeń“, które są w nim neutralizowane, jego wymiary mogą znacznie się zwiększyć, nawet do kilku centymetrów (Stevens i Lowe, 2000). Dłużej utrzymujące się powiększenie rozmiarów węzłów chłonnych nazywane jest *limfadenopatią* (łac. *lymphadenopathia*). Jest ono zazwyczaj objawem niepokojącym i wymaga szybkiej reakcji ze strony lekarza. Może być spowodowane wieloma czynnikami, z pośród których do najważniejszych można zaliczyć:

- wystąpienie stanów zapalnych towarzyszące infekcjom i związany z nimi wzrost liczby komórek odpornościowych - wówczas w węzłach produkowane są przeciwciała w odpowiedzi na pojawienie się antygeny (np. wirusa, bakterii),

- bezpośredni naciek zapalny węzłów chłonnych, występujący przy upośledzonej odporności (np. zakażenie wirusem HIV) lub w przebiegu mononukleozy,
- odczynowe powiększenie węzłów towarzyszące chorobom autoimmunologicznym, rzadziej spowodowane przyjętymi lekami, nadczynnością tarczycy czy też sarkoidozą,
- złośliwe nacieczenie węzłów przez przerzuty komórek nowotworowych lub złośliwe namnożenie komórek węzłów chłonnych (np. przy chłoniakach lub białacze).

Do węzłów chłonnych trafiają również komórki rakowe. Mogą one się oddzielać od guza pierwotnego i wnikać do limfatycznych naczyń włosowych, a następnie wraz z limfą mogą być przenoszone w inne miejsca, często dość odległe. Komórki rakowe są zazwyczaj niszczone w węzłach chłonnych, jednak czasem są w nich zatrzymywane. Rozmnażają się tam, tworząc wtórne guzy nowotworowe, tzw. przerzuty. Z tego powodu każde zauważone powiększenie węzła chłonnego, którego przyczyna nie jest znana, powinno być badane pod kątem wystąpienia komórek nowotworowych. Diagnostyka taka, może prowadzić do wykrycia zmian nowotworowych we wczesnym stadium rozwoju i tym samym może decydować o powodzeniu leczenia, a tym samym może uratować życie. Powiększone węzły chłonne w śródpiersiu są zwykle zauważane na zdjęciach rentgenowskich klatki piersiowej, dopiero gdy ich rozmiary są znaczne. Powiększenie węzłów może być również wynikiem choroby o charakterze niezłośliwym np. *sarkoidozy*, której obraz na zdjęciu rentgenowskim może przypominać chłoniaka.

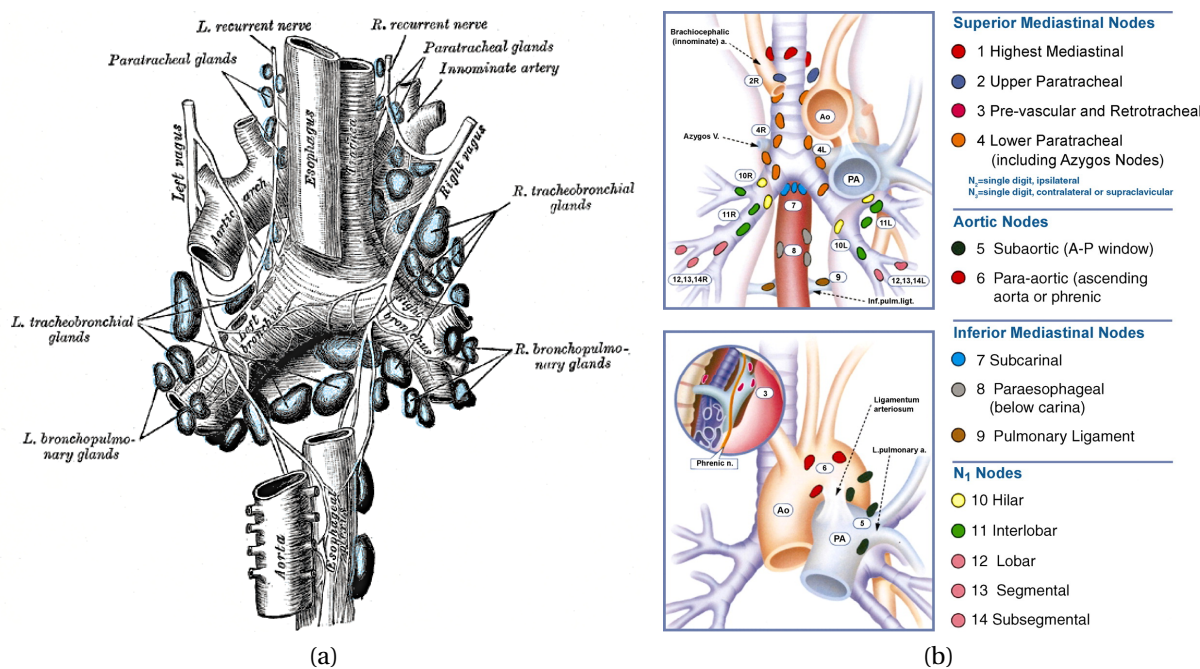
### 2.1.3 Rozmieszczenie węzłów chłonnych w klatce piersiowej

Węzły chłonne tworzą w klatce piersiowej cztery główne grupy: węzły tchawicy (ang. *tracheal*), oskrzeli (ang. *bronchial*), oskrzelowopłucne (ang. *bronchopulmonary*) oraz płucne (ang. *pulmonary*). Rysunek 2.2a przedstawia schematyczne rozmieszczenie węzłów chłonnych i innych organów w okolicach kariny. Został on opracowany na początku XX wieku przez Henrego Graya. Natomiast rysunek 2.2b został zaczerpnięty z pracy (Mountain i Dresler, 1997) i zawiera współczesną klasyfikację, rozmieszczenie oraz nazewnictwo węzłów chłonnych śródpiersia.

Do węzłów chłonnych śródpiersia według klasyfikacji Mountaina i Dreslera zaliczają się: śródpiersiowe najwyższe (gr. 1), przytchawicze górne i dolne (obustronnie) (gr. 2R, 4R, 2L, 4L), rozwidlenia tchawicy (gr. 7), okołoprzełykowe (gr. 8), okienka aortalno-płucnego (gr. 5), przednaczyńowe (gr. 3a), zatchawicze (gr. 3p), przedaortalne (gr. 6) oraz więzadła płucnego (gr. 9).

Europejska organizacja ESTS (ang. *European Society of Thoracic Surgeons*), zrzeszająca wybitnych chirurgów zajmujących się chorobami klatki piersiowej, opublikowała w 2006 roku pracę (Lardinois i inni, 2006), w której znajdują się wytyczne dotyczące zasad wykonywania diagnostyki nowotworów klatki piersiowej, w szczególności niedrobnokomórkowego raka płuc. Autorzy stwierdzają, że kluczowe znaczenie w planowaniu leczenia choroby nowotworowej jest wczesne i dokładne określenie stadium choroby nowotworowej. Jest to zadanie trudne, ponieważ wymaga określenia liczby oraz wielkości przerzutów występujących w węzłach chłonnych. Jak wynika z badań układ limfatyczny w obrębie płuc wykazuje dużą zmienność osobniczą, która skutkuje znacznymi rozbieżnościami w umiejscowieniu węzłów chłonnych (Lardinois i inni, 2003). Dlatego dużym wyzwaniem jest pobranie próbek tkanki węzłów chłonnych, kierując się jedynie wytycznymi (rysunek 2.2b) lub informacjami o ogólnej anatomii człowieka.

W pracy (Martinez-Monge i inni, 1999) przedstawiono atlas występowania węzłów chłonnych opracowany na potrzeby planowania radioterapii. Na przekrojach poprzecznych, pochodzących z tomografii komputerowej, zaznaczono obszary, w których występują węzły chłonne uzupełnione opisem ich anatomicznego położenia, nazwą, przynależnością do grupy oraz oznaczeniem skrótowym. Rysunek 2.3 przedstawia grupy węzłów chłonnych w obrębie klatki



Rysunek 2.2: Rozmieszczenie węzłów chłonnych i innych tkanek w okolicach drzewa oskrzelowego: (a) opracowane przez Graya (Gray, 1918) oraz (b) współczesna klasyfikacja Mountain/Dreslera węzłów chłonnych (Mountain i Dresler, 1997)

piersiowej zobrazowane na obrazach z tomografii komputerowej. Wyjaśnienie skrótów, wraz z obszernym opisem, znajduje się w pracy (Martinez-Monge i inni, 1999). W celu poprawienia percepcji odległości między przekrojami, na rysunku 2.4 zaznaczono numery przekroi poprzecznych z rysunku 2.3, przedstawione w płaszczyźnie czołowej.

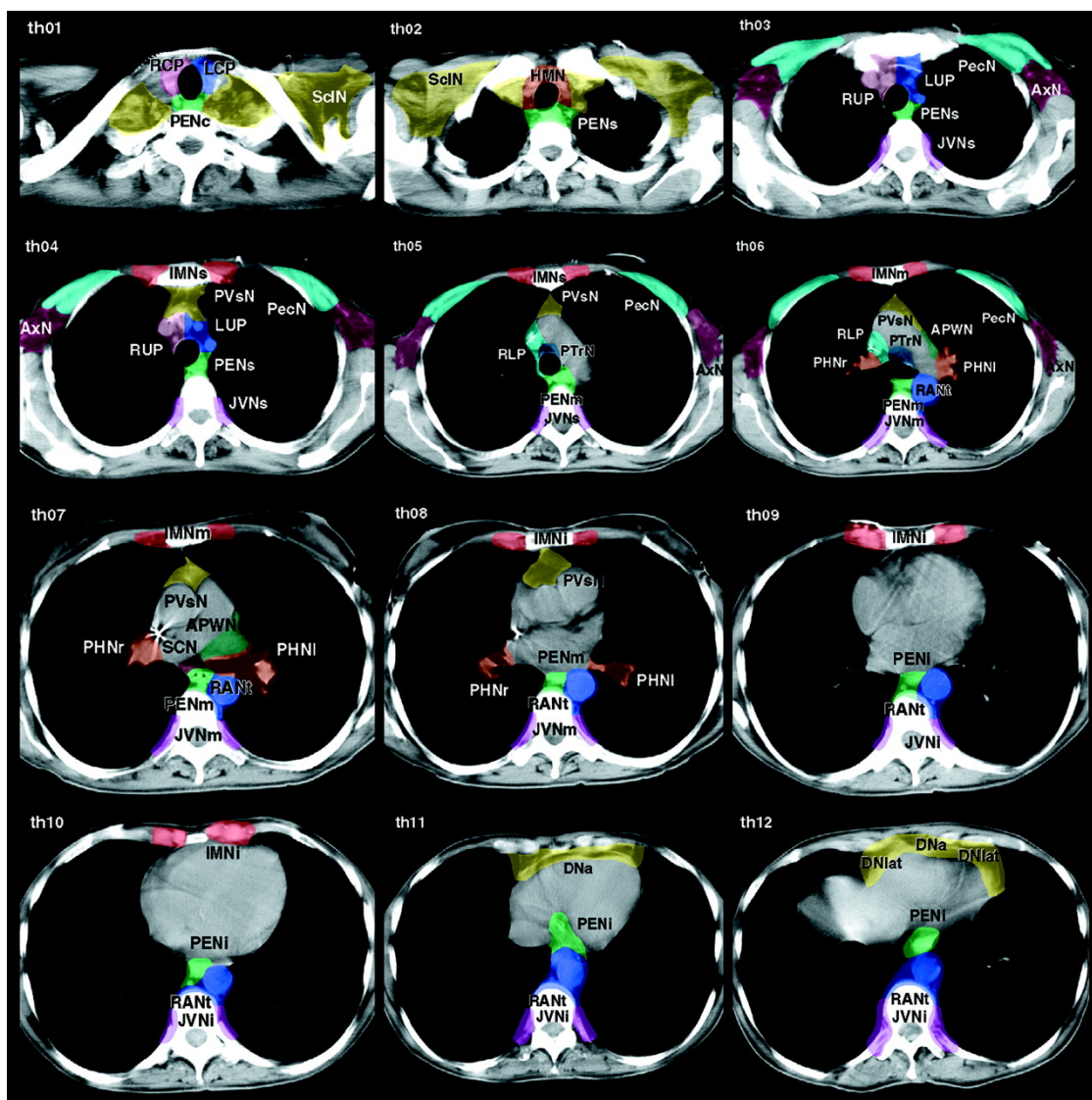
Można zauważyć, że na całej długości tchawicy (od *th01* do *th07*) znajdują się liczne grupy węzłów. Niestety, usytuowanie konkretnych węzłów chłonnych dla konkretnego pacjenta może się znacznie różnić. Dlatego też, gdy pacjent ma zdiagnozowane powiększone węzły chłonne, np. na podstawie zdjęcia rentgenowskiego, wykonywane jest dodatkowe badanie obrazowe, zazwyczaj tomografia komputerowa, umożliwiające uzyskanie serii przekroi poprzecznych klatki piersiowej. Na ich podstawie planowana jest dalsza diagnostyka.

## 2.2 Bronchoskopia

Bronchoskopia jest jedną z podstawowych, względnie małoinwazyjnych metod diagnostycznych stosowanych w pulmonologii (Pirożyński, 2011). Przede wszystkim umożliwia wzrokową ocenę (wziernikowanie) strun głosowych, krtani, ścian tchawicy i oskrzeli oraz pobranie próbek tkanki lub wydzielin, np. do badania histopatologicznego, bakteriologicznego lub cytologicznego. Dodatkowo pozwala na miejscowe podanie leków, odessanie zalegających wydzielin, usunięcie ciała obcego lub udrożnienie oskrzeli. Zabieg bronchoskopii polega na wprowadzeniu endoskopu do drzewa oskrzelowego pacjenta przez jamę nosową, plastikowy ustnik trzymany w zębach lub rzadziej przez otwór tracheotomijny.

Wyróżnia się dwa typy endoskopów stosowanych w bronchoskopii: bronchoskopy sztywne oraz bronchofibroskopy w postaci giętkiej rurki. Bronchoskop sztywny był pierwszym urządzeniem, które umożliwiało wgląd do tchawicy i oskrzeli głównych. Zbudowany jest z pro-



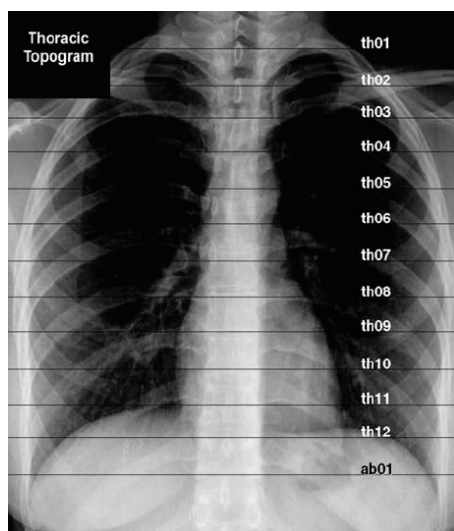


Rysunek 2.3: Obrazy CT klatki piersiowej z zaznaczonymi węzłami chłonnymi (Martinez-Monge i inni, 1999)

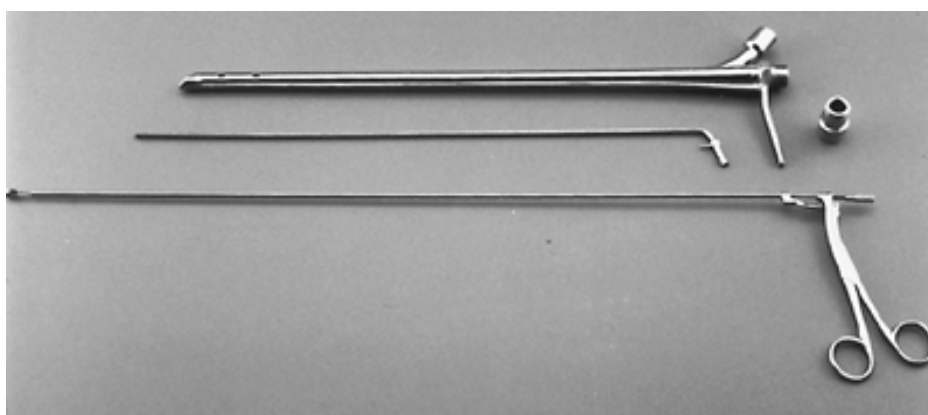
stego odcinka metalowej rurki, przez którą można włożyć układ optyczny, oświetlenie oraz niezbędne do zabiegu narzędzia (np. szczypczyki lub szczołeczkę do pobrania próbek tkanki). Zastosowanie układu optycznego soczewek z pryzmatami umożliwia obserwację ścian tchawicy oraz oskrzeli głównych. Rysunek 2.5 przedstawia przykładowy wygląd endoskopu sztywnego wraz z podstawowym osprzętem.

Zabieg bronchoskopii sztywnej wykonywany jest zazwyczaj w znieczuleniu ogólnym i stosowany jest jedynie w wyjątkowych sytuacjach, np. gdy należy przywrócić drożność drzewa tchawiczo-oskrzelowego po ciężkim urazie. Zabieg taki jest mocno obciążający dla pacjenta (konieczność wykonania pełnej anestezji), ponadto ze względu na sztywność przyrządu istnieje ryzyko uszkodzenia dziąseł, zębów lub krtani, spowodowanie krwawienia z dróg oddechowych lub mechaniczne uszkodzenie tchawicy.

Bronchofiberoskopia, czyli zabieg bronchoskopii wykonywany „giętkim” endoskopem, ma wiele zalet bronchoskopii sztywnej, jednocześnie znacznie zmniejsza ryzyko powikłań oraz zwiększa komfort pacjenta. Ponadto, dzięki mniejszej średnicy endoskopu oraz możliwości



Rysunek 2.4: Rozmieszczenie przekroi CT z rysunku 2.3 (Martinez-Monge i inni, 1999)



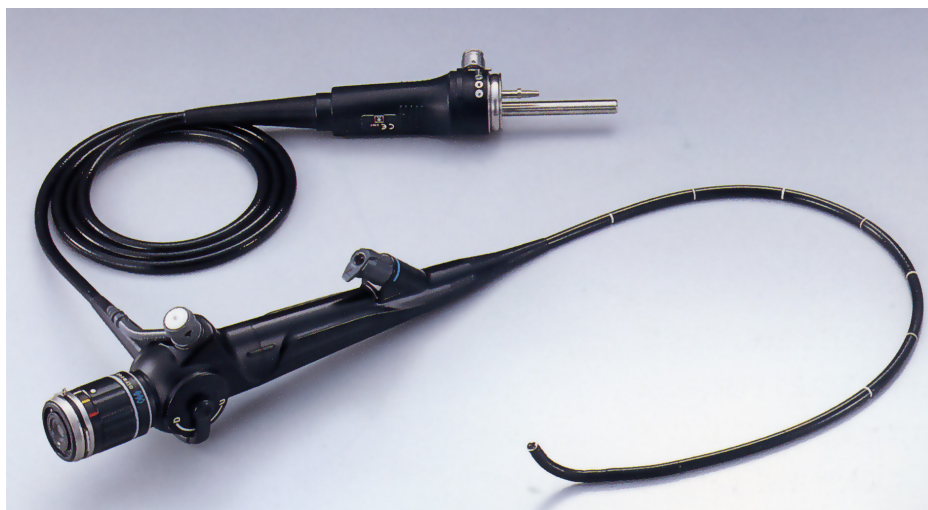
Rysunek 2.5: Przykład endoskopu sztywnego wraz z osprzętem: okulem optycznym, oświetleniem oraz szczypcami (Shields i inni, 2004)

zdalnego uginania końcówki bronchofibroskopu, możliwa jest wzrokowa inspekcja większego obszaru drzewa oskrzelowego: oskrzeli dużych, płatowych oraz segmentowych. Skonstruowanie bronchofibroskopu możliwe było dzięki wynalezieniu światłowodu.

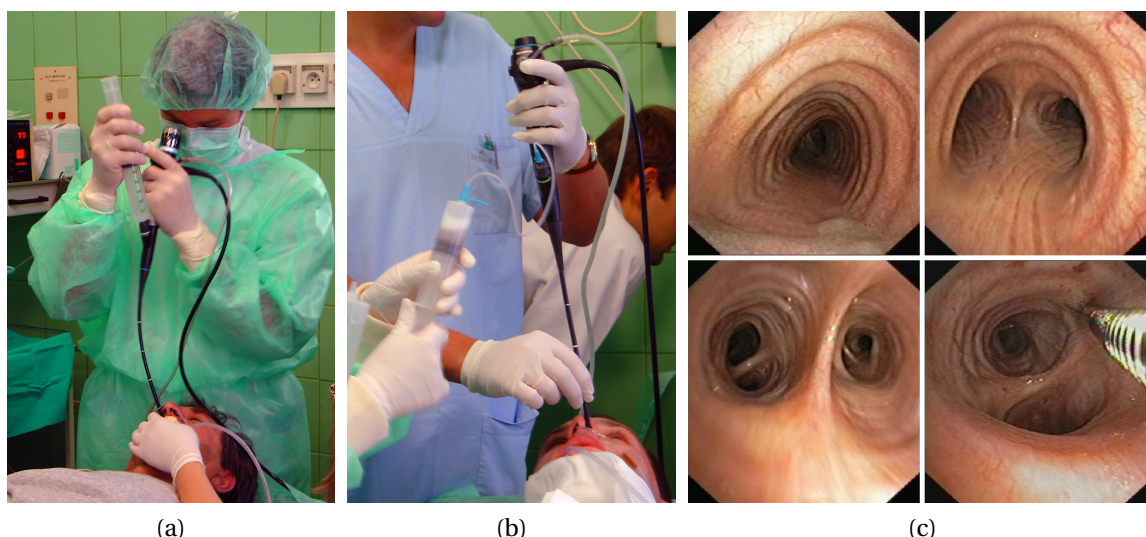
Bronchofibroskop zbudowany jest z głowicy, trzymanej w ręce przez lekarza, oraz giętkiej rurki umieszczonej w ciele pacjenta (rysunek 2.6). Na głowicy znajdują się złącza oświetlenia, okular układu optycznego lub kamera wideo, manetka (pokrętło lub dźwignia) do uginania końcówki oraz wejścia do kanału roboczego. Kanał roboczy umożliwia wprowadzenie do drzewa oskrzelowego narzędzi, takich jak: cewnik umożliwiający odessanie wydzielin, szczoteczka, igła biopsyjna, szczypczyki lub inne.

Rysunek 2.7 przedstawia przykład zabiegu bronchoskopii wykonywanej bronchofibroskopem z okulem optycznym 2.7a oraz wideobronchofibroskopem 2.7b, który w głowicy posiada wbudowaną kamerę wideo. Przykładowe obrazy zarejestrowane przy pomocy wideobronchofibroskopu przedstawiono na rysunku 2.7c.

Na rysunku 2.8 schematycznie przedstawiono podstawowe przykłady użycia bronchofibroskopu do: usunięcia ciała obcego, pobrania próbek tkanek ze ścian drzewa oskrzelowego za pomocą szczoteczki i szczypczyków. Ostatnim z prezentowanych zabiegów jest wykonanie



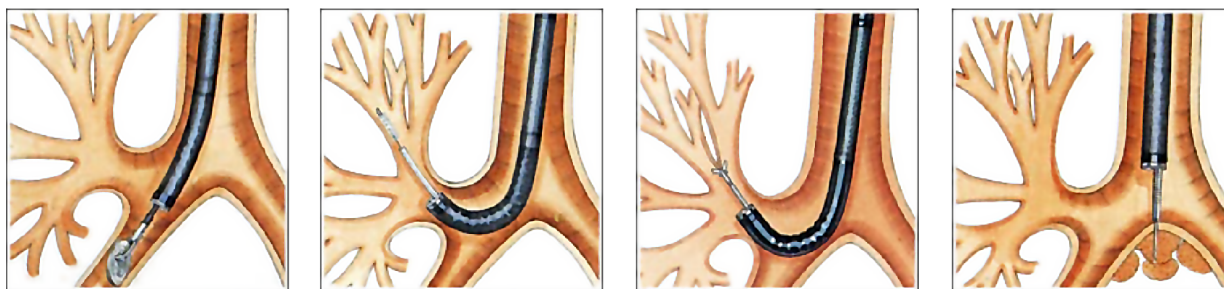
Rysunek 2.6: Bronchofiberoskop z okularzem optycznym firmy Olympus (Olimpus, 2011)



Rysunek 2.7: Zabieg bronchofiberoskopii wykonywany bronchofiberoskopem optycznym (a), wideobronchofiberoskopem wyposażonym w kamerę wideo (b) oraz przykłady obrazów drzewa oskrzelowego (c) obserwowane przez lekarza podczas bronchoskopii

*biopsji aspiracyjnej cienkoigłowej (BAC)*, podczas której pobierane są tkanki znajdujące się za ścianą drzewa oskrzelowego. W trakcie tego zabiegu, po wbiciu igły biopsyjnej w ścianę drzewa oskrzelowego, następuje zassanie przez specjalny boczny otwór igły próbki tkanek. Głównym celem wykonywania BAC jest pobranie próbek z węzłów chłonnych. Ponieważ zazwyczaj nie jest możliwe dokładne określenie położenia węzłów chłonnych za ścianą drzewa oskrzelowego, zabieg BAC powtarzany jest w konkretnym miejscu wielokrotnie do momentu, aż pobrane tkanki mogą zostać uznane za tkanki węzłów chłonnych. Jeżeli konkretny węzeł chłonny nie zostanie „odnaleziony” lekarz bronchoskopista, wykonujący badanie, przechodzi do kolejnego miejsca, w którym może znajdować się węzeł chłonny. Pobrane podczas biopsji tkanki trafiają do badania histopatologicznego.

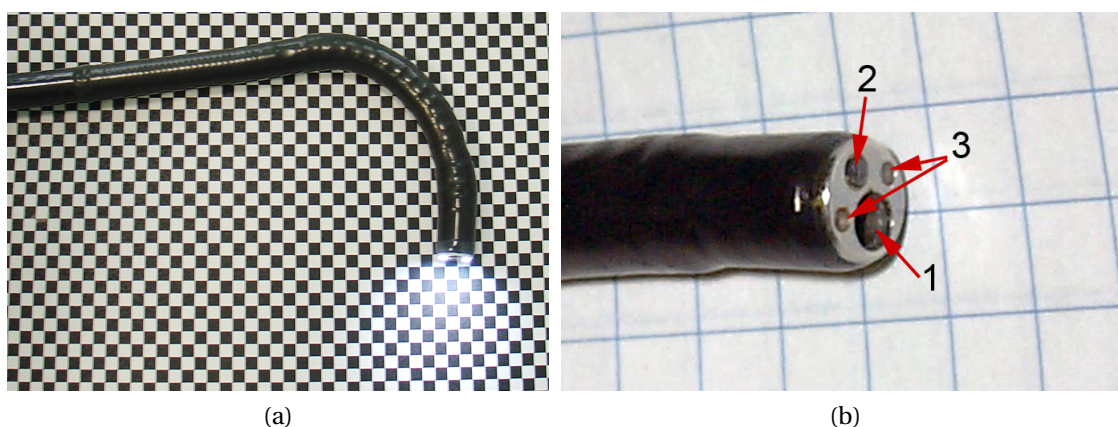




Rysunek 2.8: Przykład użycia bronchofibroskopu do: usunięcia ciała obcego, pobrania za pomocą szczoteczki próbek tkanki i wydzielin, pobrania wycinka ściany oskrzela oraz wykonania *biopsji aspiracyjnej cienkoigłowej* (BAC) (Olimpus, 2011)

Średnica zewnętrzna bronchofibroskopu waha się od 2,7mm do 6,2mm przy kanale roboczym o średnicy odpowiednio od 1,2mm do 3,2mm (Shields i inni, 2004). Endoskopy o najmniejszych średnicach stosowane są zazwyczaj w pediatrii. Zakres ugięcia końcówki endoskopu wynosi zazwyczaj 180° „do góry” oraz 130° „do dołu”. Rysunek 2.9a przedstawia ugiętą końcówkę bronchofibroskopu. Wideobronchoskopy zapewniają zwykle większy komfort pracy, dzięki możliwości oglądania powiększonego obrazu na monitorze, ale jednocześnie ich średnica zewnętrzna jest zazwyczaj większa w porównaniu z endoskopami bez kamery. Dlatego, do inspekcji odleglejszych obszarów drzewa oskrzelowego lub w przypadku diagnostyki dzieci, nadal stosuje się bronchoskopy bez kamery posiadające mniejszą średnicę, a tym samym zapewniające większy zasięg w drzewie oskrzelowym.

W giętkiej rurze znajdują się co najmniej dwa światłowody: jeden oświetlający przestrzeń znajdującą się bezpośrednio przed końcówką (rysunek 2.9a) oraz światłowód wizyjny, umożliwiający przekazywanie obrazu. Światłowód wizyjny zbudowany jest z tysięcy pojedynczych włókien światłowodowych, które odpowiednio ułożone umożliwiają przekazanie obrazu z czoła końcówki do głowicy endoskopu. Końcówka bronchofibroskopu została przedstawiona na rysunku 2.9b. Można na niej zauważyć wyjście kanału roboczego (1) soczewkę skupiającą obraz na światłowodzie wizyjnym (2) oraz dwa światłowody oświetlające (3).



Rysunek 2.9: Ugięta końcówka bronchofibroskopu (a) oraz powiększony widok czoła bronchofibroskopu (b), na którym zaznaczono: 1 – kanał roboczy, 2 – soczewkę układu optycznego oraz 3 – światłowodowe oświetlenie



Przygotowanie pacjenta do badania bronchofiberoskopowego obejmuje miejscowe znieczulenie górnych dróg oddechowych. Lekarz wykonujący badanie ma możliwość dodatkowego znieczulenia dróg oddechowych poprzez podanie środka znieczulającego przez kanał roboczy fiberoskopu. W trakcie zabiegu pacjentowi podawane są leki zmniejszające nieprzyjemne doznania związane z badaniem. Czas trwania bronchofiberoskopii zależy od celu badania i stosowanych w jego trakcie procedur pobierania materiału. Może się on wahać od kilku minut do godziny.

Pole widzenia w bronchofiberoskopie jest bardzo szerokie i może sięgać 120°. Jest to możliwe dzięki zastosowaniu soczewki szerokokątnej, która umożliwia obserwację rozległego obszaru dróg oddechowych. Wadą zastosowania optyki szerokokątnej są znaczne zniekształcenia radialne obrazu (Socha i inni, 2004). Zniekształcenia te objawiają się zakrzywianiem linii prostych na obrazie oraz problemem z poprawnym określeniem odległości oraz proporcji. Naturalny kąt widzenia dla człowieka, mierzony w płaszczyźnie poziomej, wynosi około 30°. Przy tak małym kącie widzenia nie występują zniekształcenia obrazu, pojawiające się w bronchofiberoskopie. Dlatego poprawna interpretacja obrazu z bronchofiberoskopu wymaga od lekarza dużego doświadczenia oraz przyzwyczajenia do specyfiki obrazu.

Poza bronchoskopią w pulmonologii wykonuje się również inne zabiegi umożliwiające pobranie tkanek z węzłów chłonnych śródpiersia. Można do nich zaliczyć inwazyjne zabiegi mediastinoskopii oraz tracheoskopii, w których to dostęp do węzłów chłonnych uzyskuje się poprzez rozcięcie klatki piersiowej. Przez uzyskany otwór wprowadzany jest endoskop sztywny (mediastinoskop) oraz niezbędne oprzyrządowanie. Zabiegi inwazyjne wykonywane są zazwyczaj jako ostatnie i często wiążą się usunięciem zajętych węzłów chłonnych (Dupłaga i inni, 2006).

Metodą biopsji aspiracyjnej cienkoigłowej można pobrać tkanki z węzłów pochodzących z grup: 2, 4 oraz 7 (rozdział 2.1.3). Biopsję węzłów chłonnych gr. 2 wykonuje się sporadycznie, ze względu na duże trudności techniczne związane z wygięciem końcówki bronchofiberoskopu w górnej części tchawicy. Również grupy węzłów 3, 5 i 6 znajdujące się blisko wielkich naczyń (aorty i żyły głównej) są niechętnie badane, ze względu na duże niebezpieczeństwo ich nakłucia.

## 2.3 Tomografia komputerowa

Rentgenowska tomografia komputerowa, nazywana zazwyczaj skrótowo *tomografią komputerową* TK (ang. *Computed Tomography – CT*), zrewolucjonizowała diagnostykę medyczną, dzięki możliwości nieinwazyjnego (nie licząc dawki promieniowania) obrazowania wnętrza ciała (Cierniak, 2005). Tomografia komputerowa jest *pośrednią techniką obrazowania*, w której obraz wewnętrznych struktur otrzymuje się poprzez matematyczną obróbkę (tzw. rekonstrukcję) danych pochodzących z detektorów promieniowania rentgenowskiego. Rejestrowane dane pomiarowe są miarą pochłaniania promieniowania rentgenowskiego przez tkanki i uzyskiwane są w wyniku wielokrotnego naświetlania tkanek pod różnymi kątami.

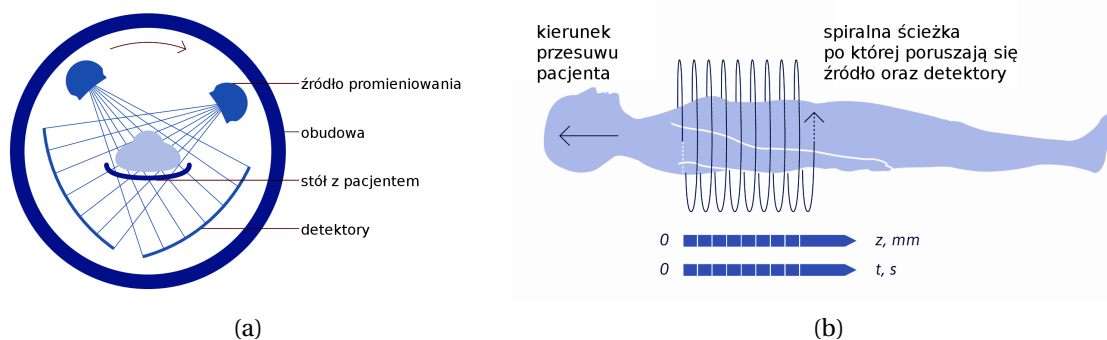
Konstrukтором pierwszego tomografu był brytyjski naukowiec Godfrey N. Hounsfield, który w pracy (Hounsfield, 1973) opisał działanie opracowanego w latach 1969–1971 tomografu. W swoich pracach Hounsfield bazował między innymi na teoretycznych opracowaniach Allana M. Cormacka. Za wynalezienie oraz skonstruowanie „tomografu wspomaganego komputerowo” (ang. *for the development of computer assisted tomography*) Godfrey N. Hounsfield wraz z Allanem M. Cormackiem otrzymali w 1979 roku Nagrodę Nobla w dziedzinie fizjologii

i medycyny. Obecnie tomografia komputerowa jest jednym z podstawowych badań diagnostycznych, szeroko stosowanym zwłaszcza w onkologii.

### 2.3.1 Zasada działania tomografii komputerowej

Działanie rentgenowskiego tomografu komputerowego opiera się na założeniu, że możliwe jest odtworzenie obrazu dwu- lub trójwymiarowego obiektu na podstawie projekcji (rzutów) badanego obiektu. Założenie to zostało udowodnione w 1917 roku przez austriackiego matematyka Johanna Radona.

Do odtworzenia wewnętrznej budowy ciała pacjenta, podczas tomografii komputerowej, wykorzystuje się promieniowanie rentgenowskie (promieniowanie X). Promieniowanie to przechodząc przez tkanki zostaje częściowo pochłonięte oraz rozproszone — zmiany te są rejestrowane i stanowią główną informację pomiarową. Źródłem promieniowania jest lampa rentgenowska, która wraz z detektorami promieniowania porusza się wokół ciała pacjenta w płaszczyźnie prostopadłej do osi ciała (rysunek 2.10a). Ruch źródła promieniowania oraz detektorów umożliwia rejestrację kolejnych projekcji wybranego fragmentu ciała. W starszych konstrukcjach ruch pacjenta względem źródła i detektorów odbywał się skokowo — każdy przekrój skanowany był oddzielnie. Współcześnie stosuje się tzw. skanowanie spiralne (rysunek 2.10b), w którym to pacjent przesuwany jest przez otwór w obudowie (gantrze tomografu) w sposób jednostajny. W tym samym czasie zespół skanujący (źródło promieniowania wraz z detektorami) obracany jest w obudowie ze stałą, zadaną prędkością obrotową. Złożenie tych dwóch ruchów daje spiralną trajektorię źródła promieniowania i detektorów względem pacjenta. Wykorzystanie detektorów wielorzędowych umożliwia zebranie wystarczającej liczby dwuwymiarowych rzutów do przestrzennej (trójwymiarowej) rekonstrukcji wnętrza ciała (Cierniak, 2005).



Rysunek 2.10: Schemat działania spiralnej tomografii komputerowej: a) wzajemne położenie pacjenta, źródła promieniowania i detektorów oraz b) spiralna ścieżka, po której poruszają się zespół źródło-detektory względem pacjenta (Siemens AG Medical Solutions, 2011)

Na podstawie zbioru dwuwymiarowych projekcji, uzupełnionego informacją o odpowiadającej im pozycji zestawu źródło-detektor, możliwa jest przestrzenna rekonstrukcja wybranego fragmentu ciała pacjenta (Kak i Slaney, 2001; Cierniak, 2005). Do najczęściej stosowanych metod rekonstrukcji obrazów z danych pomiarowych można zaliczyć metody bazujące na transformacie Fouriera i teorii przekroju Fouriera, a zwłaszcza ich modyfikację: *filtrowaną projekcję wsteczną* (Cierniak, 2005). Zagadnienia związane z teorią rekonstrukcji zostały opisane w pracy (Bułat i inni, 2006).

### 2.3.2 Interpretacja danych

W zależności od przyjętej metody rekonstrukcji oraz jej parametrów, w wyniku badania TK powstaje zbiór dwuwymiarowych obrazów, które przedstawiają serię przekroi przez ciało pacjenta. Minimalna odległość między pikselami obrazów (przekroi) głównie zależy od możliwości technologicznych tomografu. Natomiast odległość między rekonstruowanymi przekrojami ograniczona jest przede wszystkim możliwościami technicznymi aparatu, a także dopuszczalną dawką promieniowania, na jaką pacjent może być narażony w konkretnej procedurze diagnostycznej. Z jednych danych pomiarowych (jednego skanowania) możliwa jest rekonstrukcja zarówno przekroi odległych o kilka centymetrów (tzw. seria przeglądowa), jak i bardzo dokładnych przekroi o minimalnej odległości rzędu milimetrów. Skanowanie z większą rozdzielczością przestrzenną wiąże się zazwyczaj ze znacznym zwiększeniem dawki promieniowania. Najnowocześniejsze wielorzędowe tomografy są w stanie dostarczać zrekonstruowane dane izotropowe, w których odległość między pikselami oraz między warstwami jest jednokowa i może wynosić ułamek milimetra 0,24mm (Siemens, 2011).

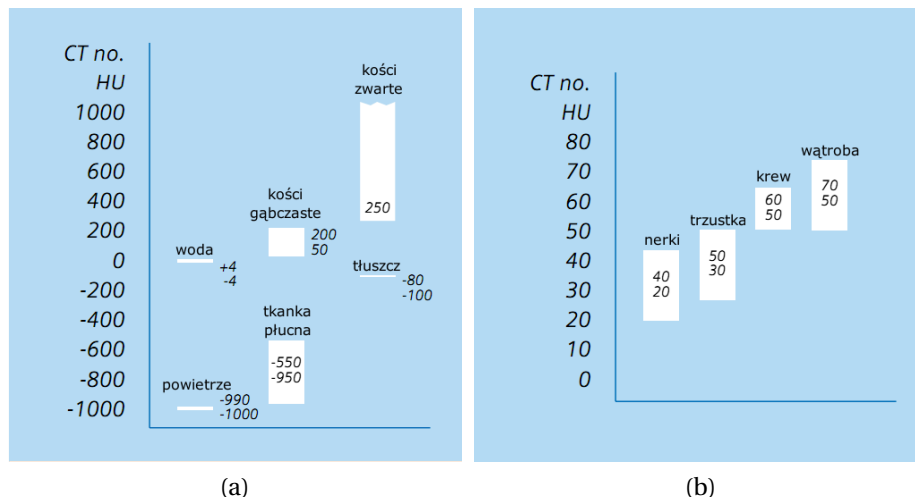
Do wyrażania stopnia osłabienia promieniowania rentgenowskiego tkanek skanowanych podczas tomografii komputerowej stosuje się normalizację zwaną *skalą Hounsfielda*. W skali tej wyrażone są wartości pikseli zrekonstruowanych obrazów. Skalowanie zmierzonych wartości osłabienia promieniowania tkanek do jednostek Hounsfielda (ang. *Hounsfield Units – HU*) odbywa się według następującej zależności:

$$1HU = K \frac{\mu - \mu_w}{\mu_w}, \quad (2.1)$$

gdzie:  $K$  jest stałą tomografu (odpowiadającą wzmocnieniu),  $\mu$  zmierzoną wartością osłabienia promieniowania, zaś  $\mu_w$  jest wartością odniesienia, równą wartości osłabienia promieniowania czystej wody. Współczynnik wzmocnienia  $K$  skalowany jest w taki sposób, by wartości  $-1000HU$  odpowiadało powietrze, zaś  $0HU$  wodzie (skalowanie liniowe, dwupunktowe).

Na rysunku 2.11 przedstawiono orientacyjne zakresy wartości danych z tomografii komputerowej wyskalowane w jednostkach Hounsfielda, z zaznaczonymi występującymi tkankami dla szerokiego zakresu wartości HU (a) oraz dla węższego zakresu (b) dla wybranych organów wewnętrznych o „miękkich” tkankach.

Zakres możliwych do osiągnięcia wartości mierzonego pochłaniania promieniowania wyrażonego w skali HU jest bardzo szeroki i zawiera się w przedziale od wartości  $-1000HU$  dla powietrza do ponad  $4000HU$  dla najbardziej nieprzenikliwych (np. zwapniałych) kości lub innych wszczepionych materiałów, np. metalowych implantów lub plomb dentystycznych. Zmiana pochłaniania o jeden stopień w skali HU odpowiada więc zmianie o 0,1% w stosunku do zakresu powietrze-woda. Hounsfield w pracy (Hounsfield, 1973) posługiwał się mniejszym zakresem liczb ( $\pm 500$ ), jednak do prezentacji wyników skanowania na wydruku zaproponował liniowe przeskalowanie zakresu do postaci tzw. okna (ang. *window*) — skończonej liczby odcieni szarości. Szerokość okna określa zakres prezentowanych danych w jednostkach HU. Najmniejsza wartość w jednostkach HU prezentowana jest jako kolor czarny, zaś największa w kolorze białym. Wartości pośrednie prezentowane są w odcieniach szarości, które dobierane są w taki sposób, by zapewnić najlepszą percepcję zmian. Istnieją szczegółowe wytyczne dotyczące sposobu prezentacji danych radiologicznych, które można znaleźć w dokumentacji standardu DICOM (NEMA, 2009). Okno wizualizacji może być definiowane przez bezpośrednie podanie wartości minimalnej ( $H_{min}$ ) i maksymalnej ( $H_{max}$ ) zakresu wartości HU lub poprzez zdefiniowanie wartości średniej, tzw. *poziomu* oznaczanego zwykle literą  $C$  (ang. *center*) oraz *zakresu* — szerokości okna, oznaczanego literą  $W$ . Związki między wartością maksymalną i minimalną



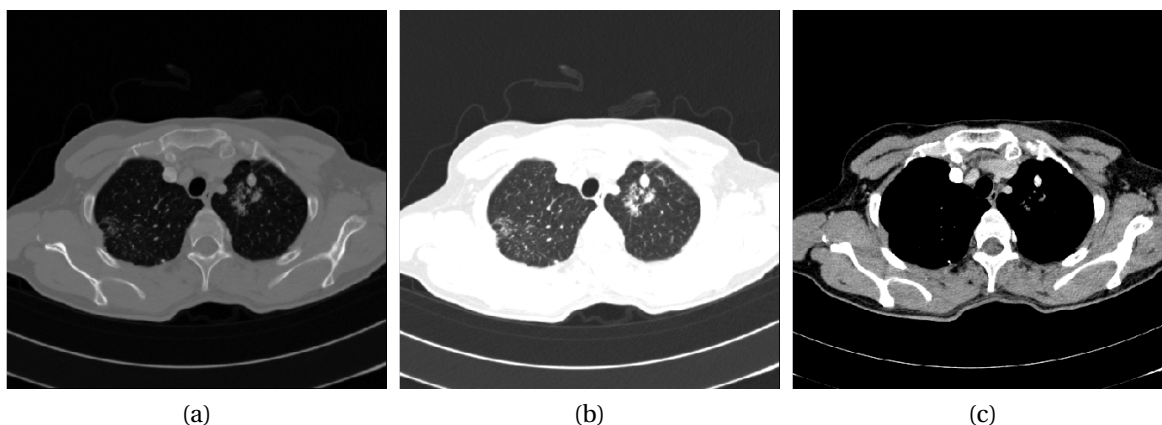
Rysunek 2.11: Skala Hounsfield: a) przykładowe zakresy wartości HU tkanek w szerokim zakresie wartości oraz b) wybrane, niskie wartości HU (Siemens AG Medical Solutions, 2011)

oraz poziomem i zakresem można zapisać w następujący sposób:

$$W = H_{max} - H_{min}, \quad (2.2)$$

$$C = \frac{W}{2}. \quad (2.3)$$

Na rysunku 2.12 przedstawiono ten sam fragment drzewa oskrzelowego, uzyskany podczas tomografii komputerowej, prezentowany w odcieniach szarości, którym przypisano różne zakresy wartości HU. Poprzez dobór poziomu oraz zakresu wizualizowanych danych możliwa jest zarówno prezentacja i analiza bardzo szerokiego zakresu wartości HU 2.12a, jak również konkretnych, wybranych tkanek 2.12b-c.



Rysunek 2.12: Fragment klatki piersiowej przedstawiony dla różnych nastaw parametrów okna  $W$  i zakresu  $C$ : (a) odcienie szarości pokrywają cały zakres danych ( $C = 217, W = 2482$ ), (b) szeroki zakres wartości prezentujący tkankę płuc ( $C = -600, W = 1200$ ) oraz (c) wąski zakres prezentowanych wartości HU ( $C = 40, W = 300$ ) dobrany do wizualizacji węzłów chłonnych

Wyniki skanowania, z przypisanymi parametrami okna wizualizacji, naświetlane są na klisze w postaci serii umieszczonych obok siebie monochromatycznych obrazów oraz archiwizacji

zowane na nośnikach cyfrowych. Klisze po wywołaniu mogą być oglądane na przeglądarkach optycznych tak, jak zwykle prześwietlenia radiologiczne (RTG). W nowoczesnych pracowniach pulmonologicznych, wyniki badań radiologicznych mogą być przeglądane na medycznych monitorach cyfrowych o poszerzonej (np. 10-12 bitowej) i kalibrowanej palecie odcieni szarości (NEMA, 2009).

Ponieważ tomografia komputerowa wywodzi się bezpośrednio z radiologicznych technik prześwietlania ciała człowieka (RTG), zaproponowana przez Hounsfielda monochromatyczna skala szarości (nawiązująca do wywołanych klisz radiologicznych) oraz prezentacja wyników w postaci „przecięć” przez ciało pacjenta została przyjęta jako standard i jest do dzisiaj powszechnie stosowana. Jednak gwałtowny rozwój technik analizy i przetwarzania danych oraz wzrost możliwości technicznych zarówno tomografów, jak i komputerów umożliwił gwałtowny rozwój metod wspomaganie badań medycznych.

## 2.4 Wspomaganie zabiegów

Najważniejszym zadaniem podczas *transbronchialnej biopsji aspiracyjnej cienkoigłowej* (oznaczanej BAC lub TBNA) jest pobranie próbek tkanek z konkretnych węzłów chłonnych znajdujących się za ścianą drzewa oskrzelowego. Ponieważ obecność węzłów za ścianą nie jest zazwyczaj manifestowana we wczesnym stadium choroby nowotworowej (przy niewielkich powiększeniach węzłów chłonnych), biopsja taka jest zabiegiem szczególnie trudnym do wykonania. Miejsce wykonania biopsji wybierane jest zazwyczaj na podstawie opisu radiologicznego oraz wyników tomografii komputerowej. Dawniej serie naświetlonych przekroi przez ciało pacjenta naświetlane były na kliszach, które następnie były przeglądane przed zabiegiem przez pulmonologa na przeglądarce ściiennej. Obecnie, coraz częściej dane z tomografii komputerowej dostępne są w postaci cyfrowej, np. są zarchiwizowane na płycie CD/DVD, którą pacjent otrzymuje po badaniu TK. Dzięki temu, możliwe jest ich pełniejsze wykorzystanie, bezpośrednio w pracowni pulmonologicznej. Daje to nowe możliwości planowania oraz wspomaganie zabiegów bronchoskopii (Bułat i inni, 2010).

Efektywność zabiegu BAC zależy głównie od zdolności, doświadczenia oraz wyobraźni przestrzennej lekarza wykonującego zabieg (Haponik i inni, 1995; Dupłaga i inni, 2006). Aby pobrać próbki z tkanek poza ścianą drzewa oskrzelowego musi on zbudować w wyobraźni przestrzenny obraz drzewa oskrzelowego na podstawie serii przekroi lub na podstawie prześwietlenia RTG (dwuwymiarowej projekcji całej klatki piersiowej). Skuteczność biopsji zależy w dużej mierze od osoby wykonującej badanie. Czułość BAC waha się w bardzo szerokich granicach od 24% do 89%.

W celu poprawy efektywności oraz skuteczności zabiegu biopsji stosuje się różne metody wspomaganie.

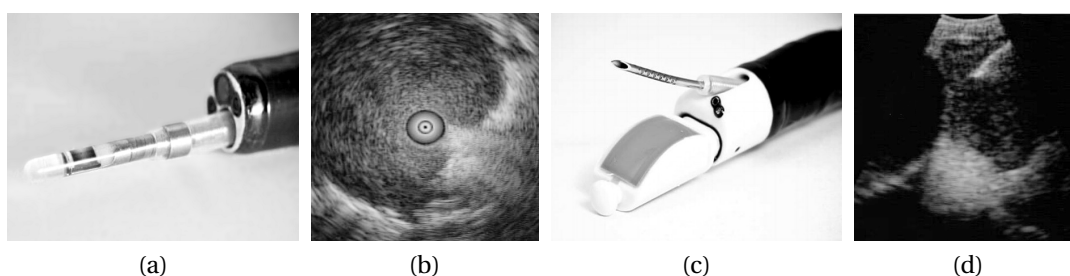
**Fluoroskopia.** Do najstarszych metod wspomaganie bezpośrednio podczas zabiegu biopsji, szczególnie wykonywanej w peryferyjnych obszarach drzewa oskrzelowego, można zaliczyć *fluoroskopię*. Jest to badanie radiograficzne, w którym „radiologiczna klisza” zastępowana jest „świecącym ekranem”. Można na nim śledzić obraz radiologiczny w czasie rzeczywistym. We współczesnych urządzeniach, miejsce „fluorescencyjnego ekranu” zajmują detektory półprzewodnikowe, np. kamera CCD. Dzięki temu, możliwe jest nagrywanie obrazu lub też przesyłanie do zdalnej konsultacji. Możliwość obserwowania pozycji endoskopu w ciele pacjenta, pozwala na wykonanie biopsji nawet w odległych obszarach drzewa oskrzelowego. Podstawową wadą

tej metody jest wykorzystywanie rentgenowskiego promieniowania jonizującego, na które bezpośrednio narażony jest pacjent oraz personel medyczny. Dawka promieniowania absorbowana przez pacjenta podczas długiego badania może być znaczna, pomimo zapewnionego minimalnego natężenia promieniowania podczas badania.

**Endobronchosonografia.** Kolejną metodą wspomaganą zabiegów BAC jest wykorzystanie ultrasonografii (Duplaga i inni, 2008). Technika wspomaganą zabiegu bronchoskopii skanowaniem USG nosi nazwę *endobronchosonografii* (ang. *endobronchial ultrasound* — *EBUS*).

Pierwsze miniaturowe sondy ultrasonograficzne zostały zaprezentowane w 1999 roku. Mogły one być wprowadzone przez kanał roboczy klasycznego bronchoskopu (rysunek 2.13a) do drzewa oskrzelowego. Dzięki temu, możliwe było uzyskanie obrazu ultrasonograficznego tkanek, znajdujących się za ścianą drzewa oskrzelowego. Miniaturowa głowica USG wiruje podczas skanowania wokół własnej osi, dzięki temu możliwe jest obrazowanie tkanek w obrębie 360° w płaszczyźnie prostopadłej do osi końcówki endoskopu (rysunek 2.13b). Ze względu na charakter pracy, sondy tego typu nazywane są również *sondami radialnymi*. W celu zapewnienia przewodności fal ultradźwiękowych między sondą i ścianą drzewa oskrzelowego, sonda USG znajduje się w elastycznej osłonie (balonie), która na czas badania wypełniana jest solą fizjologiczną.

Podczas bronchoskopii przez kanał roboczy wprowadzana jest do ciała pacjenta sonda, która po wypełnieniu osłony umożliwia skanowanie tkanek, np. w poszukiwaniu węzłów chłonnych. Po odnalezieniu pozycji węzła chłonnego, sonda jest wycofywana z kanału roboczego, a w jej miejsce umieszczana jest igła biopsyjna. Badania takiego nie można więc nazwać wspomaganym w czasie rzeczywistym. Podczas skanowania USG całe światło oskrzela jest zamykane przez sondę, blokując wymianę gazów, co można zaliczyć do wad metody. Sondy radialne pracują zazwyczaj w paśmie 20MHz, mają mały zasięg obrazowania oraz jednocześnie dużą rozdzielczość. Docelowym miejscem wykonywania biopsji wspomaganych ultrasonograficznymi sondami radialnymi są oskrzela o małej średnicy, zwłaszcza płatowe i segmentowe. Użycie sondy w oskrzelach głównych, ze względu na ich znaczne rozmiary, jest zazwyczaj bardzo utrudnione.



Rysunek 2.13: Sondy ultrasonograficzne stosowane podczas zabiegów bronchoskopii: (a) radialna sonda ultrasonograficzna wprowadzana przez kanał roboczy i (b) przykładowy obraz przez nią uzyskiwany oraz (c) głowica ultrasonograficzna zintegrowana z bronchofiberoskopem i (d) obraz przez nią uzyskiwany w czasie biopsji (Haas i inni, 2010)

Do najnowszych technik wykonywania biopsji węzłów chłonnych śródpiersia można zaliczyć zabiegi wykonywane z użyciem bronchofiberoskopu zintegrowanego z sondą ultrasonograficzną (rysunek 2.13c). System taki został wprowadzony do sprzedaży w 2004 roku pod nazwą *EBUS-TBNA* (ang. *endobronchial ultrasound-guided transbronchial aspiration*). Kosztem

pogorszenia jakości obrazu oraz ograniczenia pola widzenia, końcówka endoskopu wyposażona została w liniową głowicę USG, która umożliwia skanowanie tkanek bezpośrednio przed wylotem kanału roboczego endoskopu. Sonda ta umożliwia wykonywanie biopsji przy jednoczesnej kontroli położenia igły na obrazie ultrasonograficznym (rysunek 2.13d). Głowica USG tego typu endoskopu również wyposażona jest w elastyczną osłonę, wypełnianą w trakcie badania płynem przewodzącym fale ultradźwiękowe. Wypełniona płynem osłona jest dociskana do ściany drzewa oskrzelowego. Kąt widzenia w tego typu endoskopach jest niewielki w porównaniu z klasycznymi bronchoskopami i wynosi około 30°. Optyka endoskopu skierowana jest „na ścianę” drzewa oskrzelowego w taki sposób, by możliwe było wizualne kontrolowanie biopsji. Płaszczyzna skanowania głowicy USG przechodzi przez oś endoskopu oraz oś igły, która odchylona jest o około 20° w stosunku do osi endoskopu (rysunek 2.13c). Głowica USG pracuje z częstotliwością 7,5 MHz i obejmuje obszar 50°. Zabiegi biopsji z wykorzystaniem systemu EBUS-TBNA poprzedzane są zazwyczaj badaniem z wykorzystaniem klasycznego bronchoskopu, który zapewnia większą widoczność. Zabiegi EBUS-TBNA wykonywane są głównie w obrębie tchawicy oraz oskrzeli głównych.

Jak wynika z przeprowadzonych badań klinicznych, metody wspomaganie biopsji wykorzystujące sondy USG zwiększają szanse na zlokalizowanie powiększonych węzłów chłonnych oraz pobranie z nich próbek tkanek (Herth i inni, 2002; Duplaga i inni, 2008). Niestety, poważną przeszkodą do powszechnego ich stosowania są wysokie koszty aparatury, zwłaszcza EBUS-TBNA, oraz ograniczona żywotność sondy radialnej, która ma limitowaną liczbę „przejęć” przez kanał roboczy. Nie bez znaczenia jest również niezbędny i zazwyczaj długi czas szkolenia personelu obsługującego sondy USG.

Porównanie skuteczności biopsji wspomaganie nawigacją ultrasonograficzną (EBUS) oraz fluoroskopią dla zmian obwodowych można znaleźć między innymi w pracy (Herth i inni, 2002). Dla zmian nowotworowych o średnicy mniejszej od 3 cm ultrasonografia umożliwiła wykonanie 17 biopsji na 21 prób, zaś wspomaganie fluoroskopią 12. Daje to odpowiednio 80% dla EBUS i 57% dla fluoroskopii. Przy średnicach zmian nowotworowych ponad 3 cm, różnica była mniejsza i wynosiła 79% dla EBUS i 89% dla fluoroskopii (przy 29 próbach).

Endobronchosonografia, a w szczególności metoda EBUS-TBNA, uznawana jest za metodę o dużej skuteczności diagnostycznej i małej inwazyjności. EBUS-TBNA podobnie jak klasyczna biopsja transbronchialna umożliwia pobranie tkanek z następujących grup węzłów chłonnych (klasyfikacja Mountain/Deslera, rysunek 2.2): 2R, 2L, 3p, 4R, 4L, 7, 10R, 10L, 11R i 11L. Przyczyną trudności w obrazowaniu okolic przytchawicznych jest zazwyczaj niecałkowite przyłożenie sondy do ściany tchawicy i ujęć oskrzeli głównych oraz przyleganie węzłów chłonnych do dużych naczyń, których nakłucie mogłoby być niebezpieczne dla życia pacjenta.

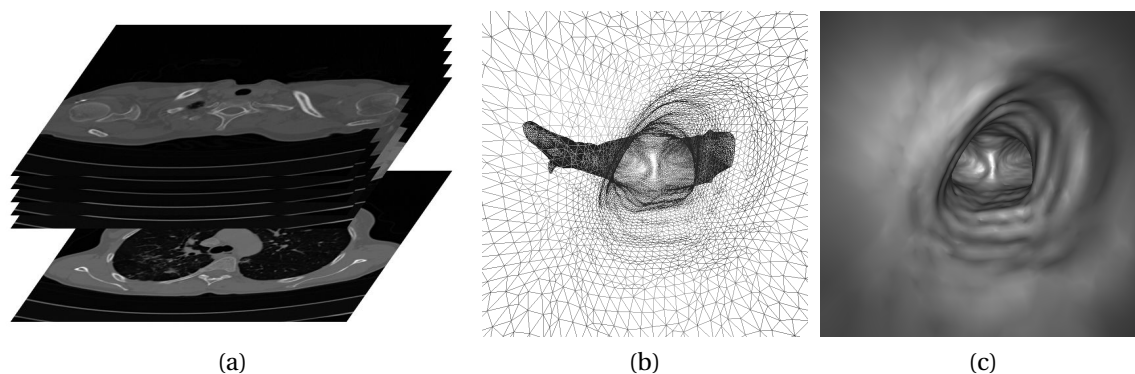
**Nowoczesne techniki obrazowania.** Coraz większy dostęp do nowoczesnych technik obrazowania, takich jak: tomografia komputerowa (CT), rezonans magnetyczny (MRI) czy też pozytonowa emisyjna tomografia komputerowa (PET), umożliwił rozwój metod wspomagających zabiegi bronchoskopii bazujących na danych pochodzących z tych systemów. Różne techniki obrazowania oferują różne możliwości diagnostyczne. Przykładowo CT i MRI oferują wysoką rozdzielczość przestrzenną, zaś PET może obrazować aktywność metaboliczną tkanek. Coraz częściej spotyka się również rozwiązania łączące różne metody obrazowania w jednym urządzeniu (np. PET-CT).

Podstawowym zastosowaniem wyników obrazowania jest analiza zeskanowanych serii dwuwymiarowych przekroi przez tkanki pacjenta pod kątem wystąpienia zmian wskazujących na procesy chorobowe, np. powiększonych węzłów chłonnych w przypadku diagnostyki no-

wotworów klatki piersiowej. Innym zastosowaniem jest używanie technik obrazowania bezpośrednio w trakcie zabiegów biopsji do kontroli położenia endoskopu oraz igły biopsyjnej względem tkanek, tak jak to ma miejsce w przypadku użycia fluoroskopii. Niewątpliwą zaletą nowoczesnych metod obrazowania jest tworzenie przestrzennych (trójwymiarowych) reprezentacji skanowanych obszarów ciała pacjenta, na podstawie których można w bardzo precyzyjny sposób określić wzajemne położenie przyrządów diagnostycznych (np. igły biopsyjnej) i tkanek.

Wyniki obrazowania mogą być również wykorzystywane do przestrzennej rekonstrukcji szczegółów anatomicznych (np. klatki piersiowej), i dzięki temu mogą ułatwić precyzyjne odnalezienie powiększonych węzłów chłonnych w wirtualnej, trójwymiarowej przestrzeni oraz mogą umożliwić zaplanowanie bronchoskopii (Bułat i inni, 2010). Możliwe jest również symulowanie zabiegów (np. bronchoskopii) jeszcze przed wykonaniem faktycznego zabiegu z udziałem pacjenta. Zabiegi takie, nazywane *wirtualnymi*, dawniej możliwe były do wykonania jedynie na oprogramowaniu aparatury skanującej, np. tomografu komputerowego, w pracowni radiologicznej. Obecnie, dzięki dużym możliwościom obliczeniowym współczesnych komputerów, możliwe są do uzyskania również w pracowni pulmonologicznej.

Ze względu na powszechną dostępność do danych z rentgenowskiej tomografii komputerowej, zostały one użyte do rekonstrukcji powierzchni drzewa oskrzelowego, wykorzystywanej w algorytmach estymacji położenia końcówki endoskopu, będących przedmiotem niniejszej pracy. Na rysunku 2.14 zaprezentowano przykład rekonstrukcji powierzchni tchawicy na podstawie danych pochodzących z tomografii komputerowej. Zrekonstruowana powierzchnia może posłużyć do wygenerowania w pełni interaktywnego obrazu wirtualnej bronchoskopii.



Rysunek 2.14: Rekonstrukcja powierzchni drzewa oskrzelowego polega na analizie serii skanów z tomografii komputerowej (a), w wyniku której powstaje przestrzenna siatka (b), która jest następnie wizualizowana (c) (Bułat i inni, 2010)

Jak już wspomniano, oprócz obrazowania wewnętrznej budowy ciała człowieka wykonywanego przed bronchoskopią, zaawansowane techniki obrazowania (CT, MRI) umożliwiają wspomaganie zabiegów biopsji w czasie rzeczywistym w trakcie ich trwania. Są to techniki wywodzące się z fluoroskopii, jednak umożliwiające uzyskanie w pełni trójwymiarowego, bieżącego obrazu ciała pacjenta wraz z aktualną pozycją endoskopu i igły biopsyjnej. W obrębie klatki piersiowej zabiegi biopsji nawigowanej tomografią komputerową (ang. *CT guided biopsy*) wykonywane są jedynie w nielicznych ośrodkach i tylko w ciężkich przypadkach zmian ułożonych obwodowo w klatce piersiowej.

Porównanie skuteczności różnych metod diagnostycznych w aspekcie określenia zaawansowania anatomicznego choroby nowotworowej (ang. *staging*) węzłów śródpiersia przedstawiono w pracy (Tolosa i inni, 2003). Zbiornicze wyniki przedstawia tabela 2.1.



Tablica 2.1: Porównanie skuteczności różnych metod diagnostycznych w określeniu anatomicznego zaawansowania choroby nowotworowej węzłów chłonnych (Toloza i inni, 2003)

Metoda wspomaganiania	Czułość %	Swoistość %	PPV %	NPV %
CT	57	82	56	83
PET	84	89	79	93
TBNA	76	96	100	71
EUS-FNA	88	91	98	77
Mediastinoskopia	81	100	100	37

PPV – wartość predykcyjna dodatnia (ang. *positive predictive value*), NPV – wartość predykcyjna ujemna (ang. *negative predictive value*), CT – rentgenowska tomografia komputerowa, PET – pozytonowa emisyjna tomografia komputerowa, TBNA – transbronchialna biopsja aspiracyjna cienkoigłowa, EUS-FNA – biopsja aspiracyjna wspierana ultrasonograficznie (ang. *endoscopic ultrasound-fine needle aspiration*)

**Systemy nawigacji optycznej.** Zabiegi endoskopowe wykonywane przyrządami sztywnymi (np. laparoskopie) od wielu lat są wspomagane systemami informatycznymi, które wykorzystują systemy optycznego śledzenia położenia instrumentów medycznych w przestrzeni (ang. *image-guided surgery*) (Preim i Bartz, 2007). Jednym z najpopularniejszych rozwiązań służących do śledzenia pozycji instrumentów są urządzenia *Polaris* firmy *NDI* lub *ARTtrack* firmy *Advanced Realtime Tracking GmbH*. Urządzenia te wykorzystują: przynajmniej dwie kamery rejestrujące obraz w podczerwieni, źródła światła podczerwonego oraz specjalne markery umieszczane na instrumentach medycznych. Położenie markerów w trójwymiarowej przestrzeni jest śledzone w czasie rzeczywistym na podstawie obrazu stereoskopowego z kamer. Rozróżnia się dwa rodzaje systemów: *aktywne*, gdy markery emitują światło (zazwyczaj diody LED) oraz *pasywne*, gdy markery (np. odbłaskowe kulki) odbijają promieniowanie z zewnętrznego oświetlacza. Różna konfiguracja przestrzenna markerów umożliwia rozróżnianie poszczególnych instrumentów medycznych. Synchronizacja otrzymywanych wyników pomiarów położenia instrumentów z przestrzenną rekonstrukcją ciała pacjenta umożliwia generowanie wirtualnego obrazu pola operacyjnego. Lekarz może obserwować na nim wzajemne położenie elementów instrumentów znajdujących się w ciele pacjenta oraz organów wewnętrznych. Ponieważ śledzenie pozycji odbywa się na podstawie obrazu markerów umieszczanych na głowicach przyrządów, muszą one być widoczne przez kamery przez cały czas badania. Dlatego systemy tego typu nie mogą być zastosowane w przypadku giętkich fiberoskopów, których pozycja końcówki nie jest zdeterminowana pozycją głowicy. Systemy te są z powodzeniem stosowane w zabiegach wykorzystujących sztywne endoskopy, takich jak zabiegi: laryngologiczne, neurologiczne, laparoskopowe czy też artroskopowe.

**Systemy nawigacji elektromagnetycznej.** Innym rozwiązaniem używanym do nawigacji w aplikacjach endoskopowych są systemy wykorzystujące fale elektromagnetyczne (ang. *electromagnetic tracking*) (Solomon i inni, 2000; Chung i inni, 2004; Mori i inni, 2005; Gildea i inni, 2006). Jednym z nich jest system *Aurora* firmy *NDI*. Zasada działania polega na wytworzeniu trzech ortogonalnych pól elektromagnetycznych przez generator umieszczony w ustalonym miejscu. W wygenerowanym polu umieszczany jest miniaturowy, wielocewkowy czujnik. Jest on podłączany do stacji bazowej (pomiarowej), która oblicza współrzędne czujnika i przesyła

współrzędne do komputera. Poza współzrędnymi obliczanymi względem generatora, wyznaczone są dodatkowo dwa lub trzy kąty obrotu sensora, co daje możliwość określenia od pięciu do sześciu stopni swobody.

Ze względu na miniaturowe wymiary sensorów, mogą być one umieszczane na końcach fiberoskopów poprzez wsunięcie przez kanał roboczy. Wadą takiego rozwiązania jest konieczność prowadzenia przewodów łączących czujnik ze stacją bazową, ograniczona żywotność czujnika przy wsuwaniu przez kanał roboczy oraz trudności z zachowaniem aseptyczności. Nie bez znaczenia są również znaczne koszty aparatury. Przykładem praktycznego wykorzystania tego typu czujników jest *superDimension i-Logic™ System*, wspierający planowanie i wykonywanie biopsji w peryferyjnych obszarach drzewa oskrzelowego (Munch, 2010).

**System wspomaganie zabiegów bronchoskopii bez dodatkowego sprzętu.** Przed wykonaniem zabiegu transbronchialnej biopsji aspiracyjnej pulmonolog dysponuje zazwyczaj danymi radiologicznymi pacjenta pochodzącymi z tomografii komputerowej, które zapisywane są w postaci cyfrowej w standardzie DICOM. Na podstawie tych danych, możliwe jest zaplanowanie zabiegu pobrania tkanek. Pod koniec lat 90-tych ubiegłego wieku zaproponowano rozwiązanie problemu lokalizacji i dotarcia do węzłów chłonnych, bazujące jedynie na danych przestrzennych pacjenta oraz obrazie wideo z endoskopu. Opisany w pracy (Bricault i inni, 1998) komputerowy system nawigacyjny wykorzystywał operację dopasowania obrazów (ang. *image registration*) polegającą na poszukiwaniu takiego położenia i kierunku patrzenia kamery wirtualnej, dla którego różnica pomiędzy obrazem wirtualnym i rzeczywistym z endoskopu była najmniejsza według przyjętej miary podobieństwa. W innych pracach można znaleźć próby rozwinięcia tej koncepcji: w pracy (Mori i inni, 2002) użyto miary korelacyjnej, zaś w pracach (Sherbondy i inni, 2000; Helferty i Higgins, 2001) entropijnej (ang. *mutual information*). Do ograniczenia złożoności poszukiwań możliwych pozycji kamery oraz w celu przyspieszenia procesu obliczeń, zaproponowano estymację położenia kamery rzeczywistej na podstawie sekwencji obrazów z endoskopu, wykorzystując do tego metody estymacji ruchu własnego (ang. *egomotion*). Można tu wymienić metody bazujące na odpowiedności punktów charakterystycznych dwóch obrazów, np. metod przepływu optycznego (ang. *optical flow*) (Helferty i Higgins, 2002; Wilson i inni, Brisbane 2005) lub metodę korelacyjną i geometrii epipolarnej (Mori i inni, 2002). W celu poprawy wyników estymacji stosowane są dodatkowe algorytmy filtracji i predykcji, np. w pracy (Nagao i inni, 2004) użyto filtra Kalmana, zaś w pracy (Deligianni i inni, 2005) filtra cząsteczkowego Monte Carlo. Poważnym ograniczeniem w stosowaniu tych metod jest stosunkowo małe zróżnicowanie elementów charakterystycznych w obrazie endoskopowym, prowadzące do trudności w automatycznym ich wyborze i śledzeniu. Jeszcze innym podejściem są metody dopasowujące struktury przestrzenne zrekonstruowanego na podstawie obrazu wideo fragmentu drzewa oskrzelowego z przestrzennym fragmentem drzewa oskrzelowego uzyskanym z tomografii komputerowej (Bartz, 2005; Deligianni i inni, 2003). Do rekonstrukcji powierzchni na podstawie obrazu używane są metody *structure-from-motion* lub *shape-from-motion* (Hartley i Zisserman, 2004).

Autor rozprawy wspólnie z lekarzem pulmonologiem oraz współpracownikami z Katedry Metrologii AGH oraz Katedry Telekomunikacji AGH od wielu lat pracują nad systemami wspomaganie zabiegów bronchoskopii, które bazowałyby jedynie na dostępnych danych: obrazie wideo rejestrowanym podczas zabiegu oraz danych z tomografii komputerowej wykonywanej przed zabiegiem bronchoskopii. Prace skupiały się na maksymalnym wykorzystaniu zasobów typowej pracowni bronchoskopowej.

W wyniku prac zespołu przebadano metody estymacji ruchu końcówki bronchoskopu bazujące na obrazie wideo. Opracowano algorytm estymacji położenia endoskopu, który został opisany między innymi w pracach (Twardowski i inni, 2006; Duda i inni, 2006). Głównym zadaniem autora niniejszej pracy było opracowanie metod szybkiego generowania wirtualnych obrazów drzewa oskrzelowego.

Do ostatnich osiągnięć interdyscyplinarnego zespołu można zaliczyć opracowanie kompleksowego systemu informatycznego BRONCHOVID wspomagającego rejestrację, archiwizację oraz analizę danych wideo gromadzonych podczas bronchoskopii (Duplaga i inni, 2010). System ten integruje również dane z tomografii komputerowej pacjenta i umożliwia wizualizację i analizę danych w celu zaplanowania zabiegu biopsji.

## MODEL ZABIEGU BRONCHOFIBEROSKOPII

Do testowania opracowanych algorytmów estymacji ruchu końcówki endoskopu niezbędne są serie obrazów testowych. Do ich generowania opracowano parametryczny model (fantom) drzewa oskrzelowego wraz z modelem kamery i oświetlenia endoskopu, które połączone razem umożliwiają generowanie obrazów wirtualnej bronchoskopii.

### 3.1 Wirtualna bronchoskopia

Dane zebrane podczas diagnostyki klatki piersiowej z wykorzystaniem rentgenowskiej tomografii komputerowej (CT) mogą posłużyć do wygenerowania obrazów tzw. wirtualnej bronchoskopii. Są to sekwencje dwuwymiarowych obrazów, które przedstawiają zrekonstruowaną na podstawie skanów oraz oświetloną powierzchnię drzewa oskrzelowego pacjenta. Dzięki temu, możliwa jest wstępna ocena kształtu ścian drzewa oskrzelowego dokonywana przez lekarza pulmonologa przed faktycznym zabiegiem bronchoskopii.

Wirtualna bronchoskopia generowana jest zazwyczaj przez oprogramowanie tomografu komputerowego w postaci statycznej sekwencji obrazów (filmu), które przedstawiają ruch wirtualnej kamery po wyznaczonej przez radiologa ścieżce. Ze względu na znaczny czas ręcznego przygotowania ścieżki przejazdu kamery, wyznaczonej przez lekarza radiologa, sekwencje wirtualnej bronchoskopii generowane na aparacie CT wykonywane były sporadycznie. Wraz ze wzrostem mocy obliczeniowej komputerów klasy PC możliwe stało się generowanie interaktywnych obrazów wirtualnych bronchoskopii poza pracownią radiologiczną, bezpośrednio przez lekarzy pulmonologów (Socha i inni, 2004). Otworzyło to nowe możliwości zarówno planowania, jak i wspomaganie zabiegów bronchoskopii.

Do wygenerowania obrazów wirtualnego drzewa oskrzelowego wykorzystuje się dane z tomografii komputerowej, które zapisywane są w standardzie DICOM, w postaci serii dwuwymiarowych przekroi. Dane z poszczególnych przekroi są „składane” do trójwymiarowej macierzy, która poza rozmiarami danych (liczbą pikseli tworzących przecięcia i liczbą warstw), uwzględnia również odległości między woksalami\* oraz współrzędne pierwszego woksela da-

---

\* trójwymiarowe odpowiedniki pikseli

nych. Dzięki temu, możliwa jest analiza anizotropowych danych (np. dokonywanie pomiarów) z zachowaniem skali.

Kolejnym krokiem generowania wirtualnej bronchoskopii jest wyznaczenie powierzchni drzewa oskrzelowego. Operacja ta sprowadza się do wyznaczenia izopowierzchni na granicy powietrze-tkanki. Jak wynika z przeprowadzonych badań (Socha i inni, 2004; Duplaga i inni, 2006) wartość progowa izopowierzchni dla drzewa oskrzelowego powinna wynosić -500HU.

Istnieje kilka technik, które umożliwiają wyznaczenie obrazu izopowierzchni w trójwymiarowych danych (Schroeder i inni, 2004; Preim i Bartz, 2007). W zależności od zastosowanej techniki wizualizacji można je podzielić na dwie główne grupy: metody bezpośrednie oraz pośrednie. Do metod bezpośrednich można zaliczyć *rendering wolumetryczny*, w którym obraz powstaje poprzez analizę trójwymiarowej macierzy danych. Przykładem tego typu algorytmu jest metoda śledzenia promieni: przez każdy piksel generowanego obrazu prowadzony jest promień, którego przecięcie z danymi (trójwymiarową macierzą danych CT) jest analizowane (Schroeder i inni, 2004). W przypadku generowania obrazu izopowierzchni analizowana jest konkretna wartość danych — jeżeli promień „trafił” w izowartość, wówczas wyznaczany jest kolor piksela rekonstruowanej powierzchni. Ze względu na duże koszty obliczeniowe, wydajność tak generowanych obrazów jest znacznie ograniczona, szczególnie przy dużej rozdzielczości tworzonego obrazu oraz znacznych rozmiarach wizualizowanych danych. Rendering wolumetryczny znajduje jednak zastosowanie w przypadku wizualizacji szerokiego zakresu wartości HU oraz w specjalistycznych trybach wizualizacji wartości maksymalnej (ang. *Maximum Intensity Projection – MIP*) lub minimalnej (ang. *Minimum Intensity Projection – MinIP*) (Preim i Bartz, 2007; Socha, 2011b).

Szybkimi metodami generowania obrazu izopowierzchni są *metody pośrednie*, do których można zaliczyć rendering oświetlonych powierzchni (ang. *Shaded-Surface Display – SSD*). Jest on realizowany dwuetapowo: w pierwszym kroku wyznaczana jest przestrzenna reprezentacja izopowierzchni, która jest w następnym kroku wydajnie wizualizowana. Ponieważ karty graficzne współczesnych komputerów są optymalizowane do wydajnego przetwarzania danych graficznych w postaci trójwymiarowych siatek trójkątów, taka reprezentacja powierzchni drzewa oskrzelowego jest jak najbardziej uzasadniona.

Jednym z najpopularniejszych algorytmów tworzenia izopowierzchni w postaci siatki trójkątów jest algorytm „kroczących kostek” (ang. *marching cubes*) (Lorensen i Cline, 1987). Algorytm ten opiera się na analizie ośmiu sąsiednich wokseli danych, które razem tworzą sześcian — każdy woxel tworzy jeden wierzchołek sześcianu. Następnie wartość każdego wierzchołka jest porównywana z zadaną wartością izopowierzchni. Jeżeli wszystkie wierzchołki sześcianu są większe lub mniejsze od zadanej wartości izopowierzchni, wówczas oznacza to, że przez ten fragment danych nie przechodzi izopowierzchnia. Jeżeli chociaż jeden wierzchołek ma inny stan niż pozostałe, oznacza to, że sześcian przecina izopowierzchnia. Na podstawie zerojedynkowych wyników porównania tworzony jest indeks komórki, który określa stan sześcianu: sposób, w jaki izopowierzchnia przecina analizowany sześcian oraz topologię połączeń trójkątów, które przybliżają izopowierzchnię. Liczba możliwych topologii jest równa  $2^8 = 256$ . Po uwzględnieniu symetrii liczbę możliwych kombinacji można ograniczyć do 15 podstawowych przypadków. W wyniku działania algorytmu dla poszczególnych komórek powstają niezależne fragmenty opisujące izopowierzchnię — przestrzenny kontur. Implementacja algorytmu jest bardzo wydajna (np. poprzez zastosowanie tablic możliwych topologii) i umożliwia wyznaczenie powierzchni w sposób równoległy, przez wiele procesorów.

Wynik działania algorytmu w postaci siatki trójkątów może być zapisany do pliku w celu ponownego użycia w przyszłości lub skierowany bezpośrednio do strumienia wizualizacji.

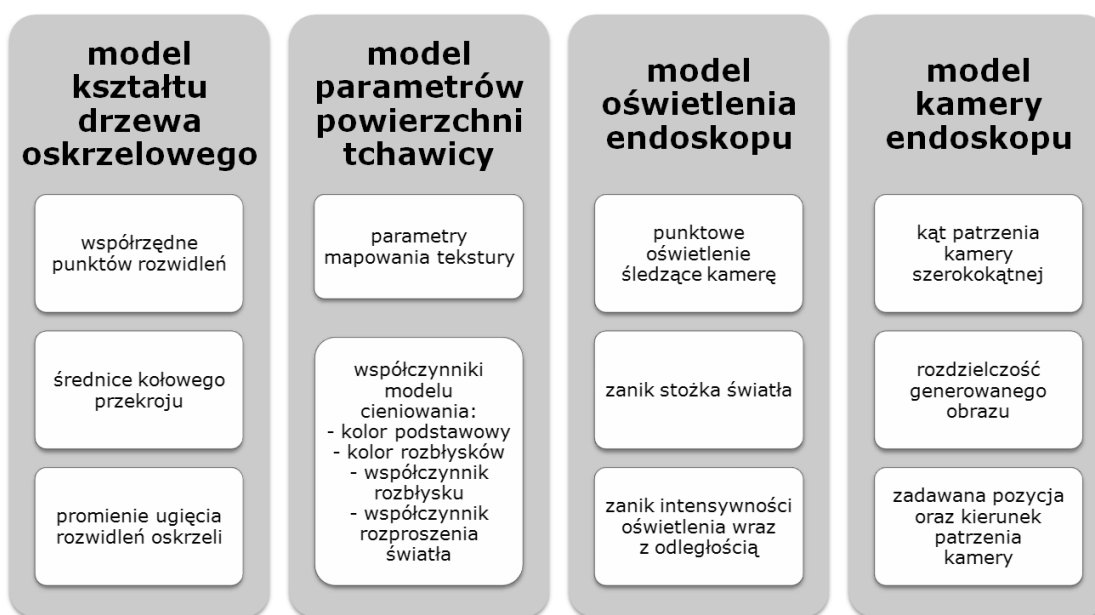
## 3.2 Model wirtualnego fantomu drzewa oskrzelowego

Podstawowym zadaniem opracowanego modelu drzewa oskrzelowego jest generowanie serii obrazów do testowania algorytmów estymacji położenia końcówki endoskopu. Model ten może być również użyty jako wirtualny fantom drzewa oskrzelowego do celów szkoleniowych.

Główne założenia wirtualnego fantomu drzewa oskrzelowego obejmują:

- możliwość tworzenie modeli o różnym stopniu skomplikowania: od bardzo prostych do szczegółowych,
- kontrola nad topologią i wymiarami drzewa oskrzelowego,
- uwzględnienie anatomicznych cech drzewa oskrzelowego, takich jak „gładkie” rozwidlenia,
- możliwość wprowadzania szczegółów anatomicznych (np. zwężeń),
- kontrola wyglądu ścian drzewa oskrzelowego,
- możliwość dodania szczegółów wyglądu tkanek (np. żyłek),
- krótki czas generowania wirtualnych obrazów.

Rysunek 3.1 przedstawia schemat blokowy poszczególnych części opracowanego przez autora rozprawy fantomu drzewa oskrzelowego, na który składają się: modele drzewa oskrzelowego (kształtu oraz wyglądu) oraz model endoskopu wraz z oświetleniem (Socha, 2011a). Poszczególne elementy tego modelu zostały szczegółowo opisane w kolejnych podrozdziałach.



Rysunek 3.1: Elementy opracowanego wirtualnego fantomu drzewa oskrzelowego i endoskopu, umożliwiające generowanie obrazów wirtualnej bronchoskopii

Opracowany parametryczny model powierzchni drzewa oskrzelowego składa się z następujących części:

- modelu kształtu początkowego odcinka drzewa oskrzelowego, który obejmuje takie parametry jak:

- współrzędne punktów rozwidleń,
- średnice kołowego przekroju,
- promienie ugięcia oskrzeli,
- modelu parametrów powierzchni tchawicy (wyglądu powierzchni), który obejmuje:
  - parametry mapowania tekstury oraz
  - współczynniki modelu cieniowania: kolor podstawowy i rozbłyśków światła, współczynniki rozbłyśku oraz rozproszenia światła.

W celu stworzenia pełnego fantomu zabiegu wirtualnej bronchoskopii model powierzchni drzewa oskrzelowego został uzupełniony o model oświetlenia i model kamery endoskopu:

- *model oświetlenia endoskopu*, w którym zastosowano punktowe oświetlenie śledzące kamerę (symulujące oświetlenie rzeczywistego endoskopu), z zanikiem stożka światła oraz zanikiem intensywności oświetlenia wraz z odległością,
- *model układu optycznego endoskopu*, którym jest szerokokątna kamera, o zadanej kątzie patrzenia, określonej rozdzielczości generowanego obrazu oraz pozycji.

Poszczególne składowe modelu drzewa oskrzelowego mogą być niezależnie parametryzowane. Umożliwia to generowanie obrazów wirtualnego drzewa oskrzelowego o różnej złożoności, dostosowanej do konkretnego eksperymentu.

Implementacja wirtualnego drzewa oskrzelowego została zrealizowana z wykorzystaniem wysokopoziomowej biblioteki *Visualization Toolkit* (Schroeder i inni, 2004; Kitware Inc., 2004), wspierającej sprzętowo możliwości generowania obrazów zaimplementowane w kartach graficznych i dostępne przez interfejs OpenGL. Do precyzyjnego modelowania kształtu oraz wyglądu tchawicy wykorzystano język GLSL (ang. *OpenGL Shading Language*), który umożliwia bezpośrednie programowanie strumienia wizualizacji OpenGL. Opis biblioteki VTK znajduje się w dodatku A.1.

### 3.2.1 Modelowanie kształtu

Podstawowym zadaniem przy modelowaniu kształtu jest stworzenie powierzchni wirtualnego drzewa oskrzelowego. Powierzchnia ta następnie służy do generowania wirtualnych obrazów bronchoskopii. Ze względu na sposób generowania obrazów, można wyróżnić dwie metody opisu drzewa oskrzelowego: w postaci przestrzennego pola skalarnego oraz w postaci siatki trójkątów. Pole skalarne można wizualizować przy pomocy kosztownego obliczeniowo renderingu objętościowego, zaś trójwymiarową siatkę można bardzo wydajnie przetwarzać z wykorzystaniem sprzętowego strumienia wizualizacji OpenGL karty graficznej.

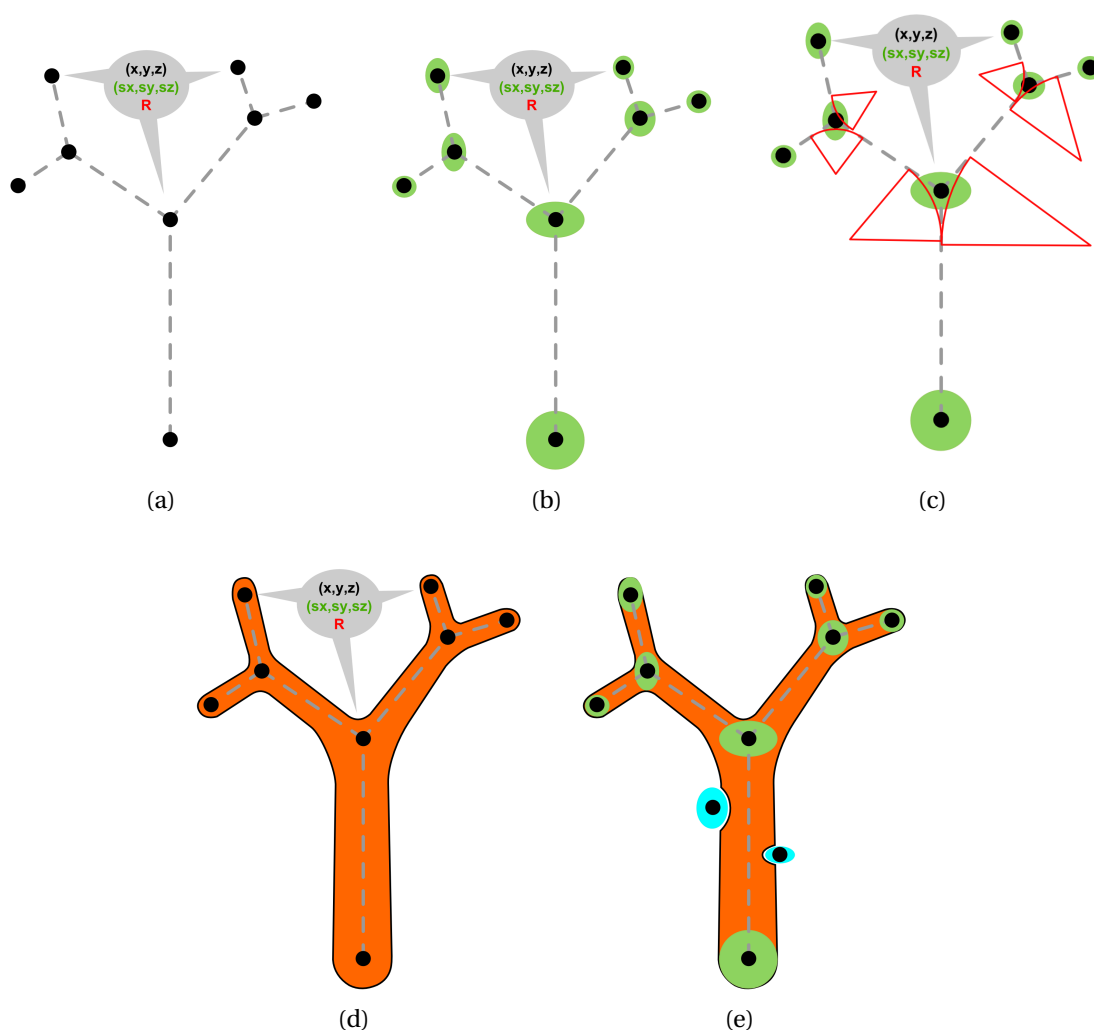
W opracowanym fantomie drzewa oskrzelowego zastosowano obydwie podejścia, wykorzystując ich najlepsze właściwości. Opis objętościowy (trójwymiarowa macierz wartości skalarne) jest bardzo zbliżony do danych medycznych pochodzących z rzeczywistych urządzeń (np. z tomografii komputerowej). Dzięki temu, w początkowej fazie modelowania możliwe jest użycie zarówno parametrycznych danych syntetycznych, jak i danych rzeczywistych, pochodzących od pacjenta. W kolejnym kroku, trójwymiarowa macierz danych jest zamieniana na trójwymiarową siatkę trójkątów, których podstawową zaletą jest możliwość wpływania na wygląd (model cieniowania, nakładanie tekstur) oraz duża wydajność generowania obrazów.

Opracowana metoda modelowania powierzchni wirtualnego drzewa oskrzelowego składa się z następujących kroków:

1. wczytanie lub wygenerowanie parametrycznego szkieletu drzewa oskrzelowego,
2. uszczegółowienie szkieletu drzewa oskrzelowego,

3. rozmieszczenie w węzłach szkieletu parametrycznych obiektów, np. przestrzennej funkcji Gaussa, których zadaniem jest „pogrubienie” szkieletu,
4. dyskretyzacja modelu: próbkowanie zadanego obszaru przestrzeni, w wyniku którego powstaje trójwymiarowa macierz wartości skalarnych,
5. wyznaczenie konturu (izopowierzchni) reprezentującej powierzchnię wirtualnego modelu drzewa oskrzelowego.

Wynikiem modelowania jest przestrzenna macierz danych (punkt 4), na podstawie której wyznaczana jest powierzchnia drzewa oskrzelowego zbudowana z siatki połączonych trójkątów (punkt 5). Dane uzyskane po dyskretyzacji mają podobny charakter, jak dane radiologiczne (np. z tomografii komputerowej) i mogą stanowić źródło danych do testowania algorytmów rekonstrukcji powierzchni lub segmentacji.



Rysunek 3.2: Kolejne etapy modelowania kształtu drzewa oskrzelowego: (a) szkielet drzewa oskrzelowego, (b) pogrubienie węzłów, (c) modelowanie łuków rozwidleń, (d) schematyczny wynik końcowy oraz (e) przykład użycia ujemnej średnicy do modelowania przewężeń

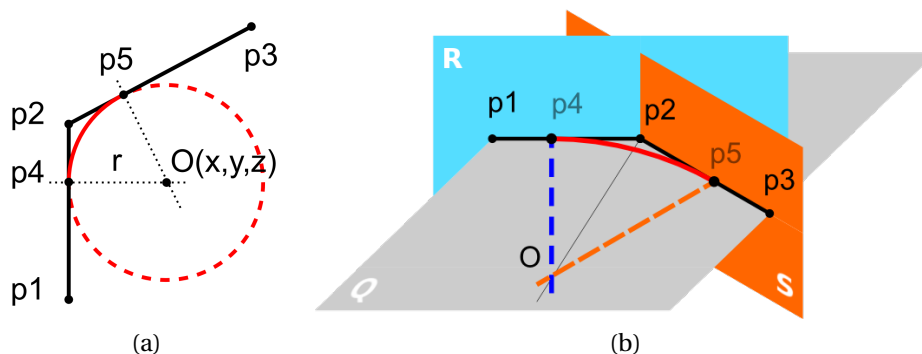


### 3.2.1.1 Szkielet drzewa oskrzelowego

Głównym celem tworzenia szkieletu jest określenie początkowego kształtu drzewa oskrzelowego. Szkielet składa się z połączonych w trójwymiarowej przestrzeni punktów, które wyznaczają miejsca rozgałęzień (rysunek 3.2a). Dodatkowe, niepołączone punkty w przestrzeni, służą do modelowania zniekształceń powierzchni (np. lokalnych zwężeń, rysunek 3.2e). Do każdego punktu szkieletu przypisywana są dwie wartości: wektorowa, która służy do określenia średnicy przekroju poprzecznego oskrzeli oraz wartość skalarna, parametryzująca promień łuku rozgałęzienia. Punkty, przypisane im wartości liczbowe oraz połączenia przechowywane są jako dane w obiekcie typu *vtkPolyData*. Początkowa wersja geometrii drzewa oskrzelowego (szkielet), złożona zazwyczaj z niewielkiej liczby punktów, wymaga dalszego uszczegółowienia. Podstawowym problemem występującym przy tak prostym opisie są niefizjologiczne rozwidlenia, które wymagają wygładzenia.

### 3.2.1.2 Dodanie szczegółów

Występujące w rzeczywistym drzewie oskrzelowym rozwidlenia mają charakter gładkich przejść, które można zamodelować w postaci wycinka okręgu (Liu i inni, 2003). W tym celu, dla każdego wierzchołka szkieletu drzewa oskrzelowego, połączonego z innymi wierzchołkami, wyznaczane są współrzędne nowych punktów, które tworzą łuk o zadanym promieniu. Na rysunku 3.3 przedstawiono konstrukcję łuku wygładzającego rozwidlenie drzewa oskrzelowego zawarte między punktami  $p_1$ ,  $p_2$  oraz  $p_3$ . Punkt  $p_2$  jest wierzchołkiem szkieletu drzewa oskrzelowego, który w wyniku obliczeń zostanie zastąpiony łukiem.



Rysunek 3.3: Konstrukcja łuku o zadanym promieniu  $r$ , wygładzającego rozgałęzienie wyznaczone przez punkty  $p_1 - p_2 - p_3$ : (a) na płaszczyźnie oraz (b) w przestrzeni, z zaznaczonymi płaszczyznami prostopadłymi do ramion łuku

Można zauważyć, że trzy punkty w przestrzeni, nieleżące na jednej prostej w sposób jednoznaczny wyznaczają płaszczyznę  $Q$  (rysunek 3.3b), na której musi się znaleźć łuk „wygładzający” rozwidlenie drzewa oskrzelowego. Przy zadanym promieniu  $r$  oraz znanych współrzędnych punktów  $p_1$ ,  $p_2$  i  $p_3$  należy wyznaczyć: punkt  $O(x, y, z)$ , określający środek okręgu oraz punkt początku  $p_4$  i końca  $p_5$  łuku. Punkt  $O$  znajduje się w równej odległości od prostych  $(p_1, p_2)$  oraz  $(p_2, p_3)$ .

Na podstawie równania tych prostych można wyznaczyć równania płaszczyzn  $R$  i  $S$  do nich prostopadłych i przechodzących przez punkt  $p_2$ . Na podstawie wartości kąta przecięcia płaszczyzn  $R$  i  $S$  ( $\angle p_4Op_5$ ) oraz długości promienia  $r$  wyznaczana jest odległość  $l = |p_2 - p_4| =$

$|p_2 - p_5|$ . Następnie płaszczyzny  $R$  oraz  $S$  są przesuwane równoległe o odległość  $l$  odpowiednio do punktów  $p_4$  i  $p_5$  (linie kropkowane na rysunek 3.3b).

Równanie na współrzędne punktu  $O$  zostało wyprowadzone z przecięcia trzech płaszczyzn:  $Q$  oraz przesuniętych  $R$  i  $S$ .

**Wyprowadzenie.** Przyjęto następujące oznaczenia: współrzędne  $i$ -tego punktu  $p_i$  wyznaczone są przez  $(x_i, y_i, z_i)$ , gdzie  $i \in \{1, 2, 3\}$ .

Równanie parametryczne płaszczyzny  $Q$  wyznaczają współrzędne trzech rozpatrywanych punktów (Bronsztejn i Siemiendajew, 1973):

$$x \cdot A_Q + y \cdot B_Q + z \cdot C_Q + D_Q = 0 \quad (3.1)$$

gdzie:

$$A_Q = (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1), \quad (3.2a)$$

$$B_Q = (x_3 - x_1)(z_2 - z_1) - (y_2 - y_1)(z_3 - z_1), \quad (3.2b)$$

$$C_Q = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1), \quad (3.2c)$$

$$D_Q = -x_1 \cdot A_Q - y_1 \cdot B_Q - z_1 \cdot C_Q. \quad (3.2d)$$

Równanie płaszczyzny  $S$  prostopadłej do wektora  $\overrightarrow{p_1 p_2}$ , zapisane w postaci parametrycznej ma następującą postać:

$$\begin{aligned} & x \cdot \underbrace{(x_2 - x_1)}_{A_S} + y \cdot \underbrace{(y_2 - y_1)}_{B_S} + z \cdot \underbrace{(z_2 - z_1)}_{C_S} - \\ & - \underbrace{[x_2(x_2 - x_1) + y_2(y_2 - y_1) + z_2(z_2 - z_1)]}_{D_S} = 0 \end{aligned} \quad (3.3)$$

Analogicznie, równanie płaszczyzny  $R$ , prostopadłej do wektora  $\overrightarrow{p_2 p_3}$  opisuje następujące równanie:

$$\begin{aligned} & x \cdot \underbrace{(x_2 - x_3)}_{A_R} + y \cdot \underbrace{(y_2 - y_3)}_{B_R} + z \cdot \underbrace{(z_2 - z_3)}_{C_R} - \\ & - \underbrace{[x_2(x_2 - x_3) + y_2(y_2 - y_3) + z_2(z_2 - z_3)]}_{D_R} = 0. \end{aligned} \quad (3.4)$$

Kąt między płaszczyznami  $S$  oraz  $R$  wynosi:

$$\cos(\Phi) = \frac{A_S \cdot A_R + B_S \cdot B_R + C_S \cdot C_R}{\sqrt{[A_S^2 + B_S^2 + C_S^2] \cdot [A_R^2 + B_R^2 + C_R^2]}}, \quad (3.5)$$

$$(3.6)$$

a stąd otrzymuje się:

$$\phi = \pi - \arccos(\Phi). \quad (3.7)$$

Odległość  $l = |\overrightarrow{p_2 p_4}| = |\overrightarrow{p_2 p_5}|$  zależy od zadanego promienia łuku  $r$  oraz kąta  $\phi$ , zawartego między płaszczyznami:

$$l = r \cdot \operatorname{tg}\left(\frac{\phi}{2}\right). \quad (3.8)$$

Znając odległość  $l$  możliwe jest wyznaczenie współrzędnych punktów  $p_4$  oraz  $p_5$  (początek i koniec łuku) :

$$p_4 = \left( \frac{l \cdot (x_1 - x_2)}{|\vec{p}_1 \vec{p}_2|}, \frac{l \cdot (y_1 - y_2)}{|\vec{p}_1 \vec{p}_2|}, \frac{l \cdot (z_1 - z_2)}{|\vec{p}_1 \vec{p}_2|} \right), \quad (3.9)$$

$$p_5 = \left( \frac{l \cdot (x_3 - x_2)}{|\vec{p}_3 \vec{p}_2|}, \frac{l \cdot (y_3 - y_2)}{|\vec{p}_3 \vec{p}_2|}, \frac{l \cdot (z_3 - z_2)}{|\vec{p}_3 \vec{p}_2|} \right). \quad (3.10)$$

Kolejnym krokiem algorytmu jest przesunięcie płaszczyzn  $R$  i  $S$  do punktów  $p_4$  i  $p_5$ . W tym celu wyznaczane są nowe współczynniki kierunkowe dla równań (3.3) i (3.4):

$$D_S = -x_4 \cdot A_S - y_4 \cdot B_S - z_4 \cdot C_S, \quad (3.11)$$

$$D_R = -x_5 \cdot A_R - y_5 \cdot B_R - z_5 \cdot C_R. \quad (3.12)$$

Punkt przecięcia płaszczyzn  $Q$ ,  $R$  i  $S$  wyznacza punkt  $p_0$  (rysunek 3.3) będący środkiem okręgu wpisanego w łuk:

$$p_0 = \left( -\frac{\Delta_x}{\Delta}, -\frac{\Delta_y}{\Delta}, -\frac{\Delta_z}{\Delta} \right) \quad (3.13)$$

gdzie:

$$\Delta = A_S B_R C_Q + B_S C_R A_Q + C_S A_R B_Q - C_S B_R A_Q - B_S A_R C_Q - A_S C_R B_Q, \quad (3.14a)$$

$$\Delta_x = D_S B_R C_Q + B_S C_R D_Q + C_S D_R B_Q - C_S B_R D_Q - B_S D_R C_Q - D_S C_R B_Q, \quad (3.14b)$$

$$\Delta_y = A_S D_R C_Q + D_S C_R A_Q + C_S A_R D_Q - C_S D_R A_Q - D_S A_R C_Q - A_S C_R D_Q, \quad (3.14c)$$

$$\Delta_z = A_S B_R D_Q + B_S D_R A_Q + D_S A_R B_Q - D_S B_R A_Q - B_S A_R D_Q - A_S D_R B_Q. \quad (3.14d)$$

Znając środek okręgu  $p_0$  oraz początek i koniec łuku (punkty  $p_4$  i  $p_5$ ), możliwe jest wyznaczenie punktów tworzących łuk  $\widehat{p_4 p_5}$  poprzez obrót wektora  $\vec{p_0 p_4}$  po płaszczyźnie  $Q$  od punktu  $p_4$  do  $p_5$ , o określony kąt  $\tau$ . W tym celu, na podstawie współrzędnych punktu  $p_0$  wyznaczana jest macierz translacji  $T$ :

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.15)$$

oraz macierz obrotu  $U$ , obliczana na podstawie współrzędnych punktów  $p_0$ ,  $p_4$  oraz  $p_5$ :

$$U_Z = \frac{(p_4 - p_0)}{|\vec{p_4 p_0}|}, \quad (3.16)$$

$$U_X = \frac{(p_5 - p_0) \times (p_4 - p_0)}{|\vec{p_5 p_0} \times \vec{p_4 p_0}|}, \quad (3.17)$$

$$U_Y = U_Z \times U_X, \quad (3.18)$$

$$U = \begin{bmatrix} U_X & 0 \\ U_Y & 0 \\ U_Z & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} U_{Xx} & U_{Xy} & U_{Xz} & 0 \\ U_{Yx} & U_{Yy} & U_{Yz} & 0 \\ U_{Zx} & U_{Zy} & U_{Zz} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.19)$$

Punkt początkowy łuku  $p_4$ , jest następnie przesuwany do początku układu współrzędnych:

$$p'_4 = U \cdot T \cdot p_4 = U \cdot T \cdot \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}. \quad (3.20)$$

Dzięki przekształceniu (3.20), możliwe jest zastosowanie macierzy obrotu do wyznaczenia kolejnych punktów.

Zakładając, że łuk ma składać się z  $n$  punktów, kąt obrotu  $\tau$  wynosi:

$$\tau = -\frac{\phi}{n}, \quad (3.21)$$

gdzie  $\phi$  jest kątem rozwarcia łuku (3.7). Liczba punktów  $n$  może być również parametryzowana odległością między punktami  $dp$ , wówczas:

$$n = \frac{\phi \cdot r}{dp}. \quad (3.22)$$

Znając kąt obrotu  $\tau$ , można wyznaczyć współczynniki macierzy obrotu  $W_x$ :

$$W_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \tau & -\sin \tau & 0 \\ 0 & \sin \tau & \cos \tau & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.23)$$

Punkty leżące na łuku obliczane są w sposób iteracyjny, poprzez mnożenie współrzędnych punktu poprzedniego przez macierz obrotu  $W_x$ :

$$p'_{i+1} = W_x \cdot p'_i \quad (3.24)$$

Pierwszym obracany punkt jest przekształcony w równaniu (3.20) punkt początkowy  $p_4$ . Obliczenia powtarzane są  $n$  razy. Otrzymane w ten sposób punkty są przesuwane na płaszczyznę  $Q$  przy pomocy zależności:

$$p_i = (R \cdot T)^{-1} \cdot p'_i \quad (3.25)$$

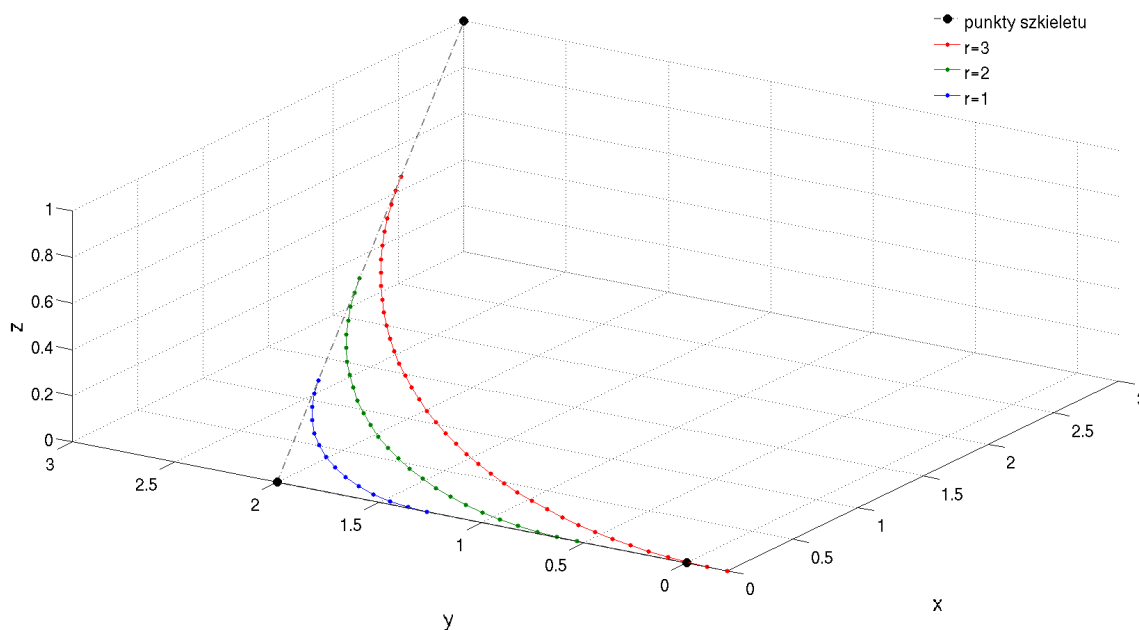
Rysunek 3.4 przedstawia przykład wygładzania rozgałęzienia drzewa oskrzelowego opisaną powyżej metodą. Dla trzech punktów wygenerowano trzy łuki o promieniach 1, 2 oraz 3. Liczba generowanych punktów była zależna od odległości między nimi zgodnie z zależnością (3.22) i wynosiła  $dp = 0,1$ .

### 3.2.1.3 Pogrubienie szkieletu

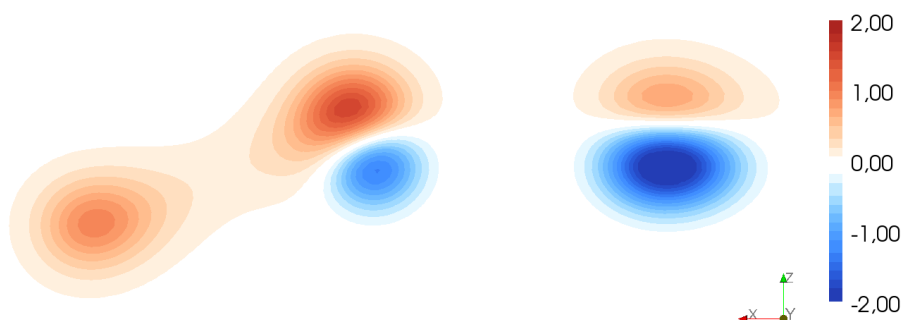
Kolejnym krokiem modelowania kształtu drzewa oskrzelowego jest „pogrubienie” szkieletu. Można to osiągnąć poprzez obliczenie transformaty odległości (Tadeusiewicz i Korohoda, 1997) dla szkieletu (filtr *vtkImplicitModeller*), otrzymując model uproszczony („tubowy”) o jednej stałej średnicy. Model ten, po zamianie na siatkę trójkątów, może być zmodyfikowany poprzez nałożenie tekstury przesuwającej wierzchołki trójkątów wzdłuż wektorów normalnych.

Modelowanie bardziej różnorodnych kształtów uzyskuje się poprzez rozmieszczenie w węzłach szkieletu parametrycznych brył. W przypadku zastosowania filtra *vtkGaussianSplatter* jest to trójwymiarowy rozkład Gaussa. Parametry rozkładu mogą być skalowane wartością wektorową, przypisaną do punktów szkieletu. Wektor ten zawiera trzy współczynniki skalujące rozkład w trzech kierunkach. Dzięki temu, możliwe jest modelowanie lokalnych zmian średnicy oraz wprowadzanie dodatkowych punktów modelujących zniekształcenia (np. zwężenia). Długość wektora określa maksymalną wartość amplitudy rozkładu. Wystąpienie ujemnego współczynnika zmienia znak amplitudy rozkładu na ujemny. Na rysunku 3.5 przedstawiono przekrój przez model drzewa oskrzelowego z zastosowanymi punktami o ujemnej i dodatniej wartości.

Użycie ujemnych wartości spowodowało „wypłaszczenie” fragmentu drzewa oskrzelowego. W taki sposób możliwe jest modelowanie przewężeń lub innych anatomicznych elementów drzewa oskrzelowego, np. zmian o charakterze guza.



Rysunek 3.4: Przykład wygładzania rozgałęzienia przez wpisanie łuku



Rysunek 3.5: Przykład dwuwymiarowego przekroju przez pole skalarne wygenerowane podczas pogrubiania szkieletu

### 3.2.2 Rekonstrukcja powierzchni

Ostatnim krokiem modelowania kształtu jest dyskretyzacja wcześniej przygotowanego opisu drzewa oskrzelowego, w wyniku której otrzymywana jest trójwymiarowa macierz wartości skalarnej (przestrzenne pole skalarne). Podstawowymi parametrami tej operacji są: przestrzenny zakres współrzędnych, obejmujący obszar, w którym rozmieszczono punkty szkieletu oraz krok dyskretyzacji. Dla wszystkich tak określonych punktów w przestrzeni obliczana jest odległość od punktów szkieletu, a następnie wyznaczana jest suma rozkładów Gaussa.

W wyniku tych obliczeń otrzymywana jest trójwymiarowa macierz wartości skalarnej, podobna do macierzy z danymi medycznymi (np. CT, MRI). Rysunek 3.6 przedstawia objętościową wizualizację przykładowego szkieletu (czarna linia) oraz uzyskanego na jego podstawie rozkładu modelującego kształt drzewa oskrzelowego.

Kolejnym etapem generowania powierzchni jest wyznaczenie izopowierzchni reprezentującej powierzchnię wirtualnego modelu drzewa oskrzelowego. W tym celu zastosowano filtr konturujący *vtkMarchingCubes*, który generuje siatkę trójkątów (Lorensen i Cline, 1987). Na ja-



Rysunek 3.6: Wizualizacja pola skalarnego wygenerowanego na podstawie szkieletu drzewa oskrzelowego

kość generowanej siatki trójkątów duży wpływ ma rozdzielczość przestrzenna, z jaką dokonywana jest dyskretyzacja. W celu poprawy jakości siatki (np. wyeliminowanie trójkątów typu „drzazga”, które mogą powodować artefakty w obrazie) zastosowano filtry decymujące *vtkDecimatePro* (Schroeder i inni, 1992) oraz wygładzające siatkę trójkątów *vtkSmoothPolyDataFilter* (Hansen i inni, 2005). Rysunek 3.7 przedstawia przykład izopowierzchni wygenerowanej w taki sposób.

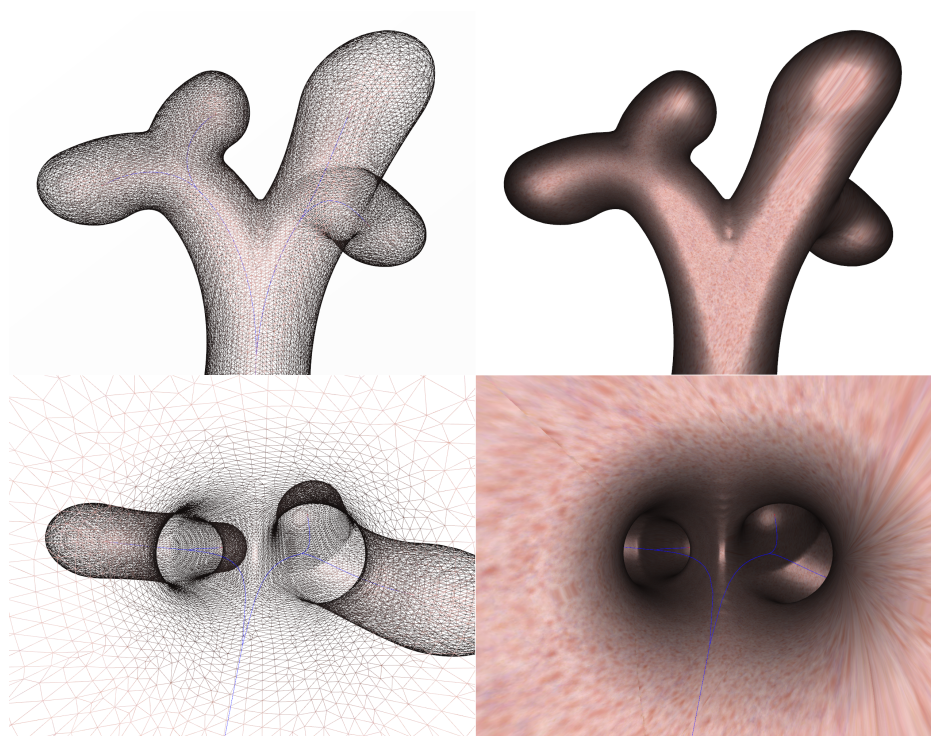
Zaprezentowane podejście łączy zalety modelowania parametrycznymi bryłami podstawowymi (duża możliwości kontroli) z łatwością i szybkością implementacji. Co ważne, przetwarzanie danych od momentu konturowania jest takie same, jak dla rzeczywistych danych medycznych (np. pochodzących z tomografii komputerowej).

### 3.2.3 Modelowanie wyglądu powierzchni

Powierzchnia drzewa oskrzelowego pokryta jest dwoma rodzajami tkanek, których głównym zadaniem jest ochrona płuc przed zanieczyszczeniami. Są to: komórki wydzielające lepki śluz i surowicę oraz komórki urzęsione, których zadaniem jest przesuwanie zanieczyszczeń w kierunku krtani. Komórki urzęsione nie są widoczne bezpośrednio w obrazie endoskopowym, zaś obecność śluzu objawia się w postaci „mokrego” błyszczącego ścian układu oddechowego. Brak błyszczącego może świadczyć o zmianach chorobowych.

Na zdrowych ścianach tchawicy i oskrzeli widoczne są naczynia krwionośne i tkanki mięśni gładkich. Wygląd powierzchni może być bardzo różny, zwłaszcza gdy występują zmiany chorobowe, takie jak: krwawienia, wybroczyny, zmiany guzowate i naciekowe oraz inne. W stanach chorobowych często obserwuje się znaczne ilości wydzieliny, która zalega oraz przemieszcza się w drzewie oskrzelowym.

W prezentowanym modelu zastosowano dwie techniki, bezpośrednio decydujące o wyglądzie wirtualnej powierzchni drzewa oskrzelowego. Pierwsza polega na automatycznym gene-



Rysunek 3.7: Wyniki rekonstrukcji modelu powierzchni drzewa oskrzelowego: siatka trójkątów opisująca powierzchnię oraz oświetlona powierzchnia

rowaniu współrzędnych mapowania tekstur. Ze względu na kształt zastosowano mapowanie cylindryczne. Dzięki temu, możliwe jest nakładanie bitmapy reprezentującej wygląd tkanek, występujących w rzeczywistym drzewie oskrzelowym.

Drugą techniką jest wykorzystanie możliwości przeprogramowania algorytmów generowania obrazu karty graficznej na potrzeby zmiany standardowych algorytmów cieniowania (Randima i Kilgard, 2003). Programy dla jednostek cieniujących znajdują się w oddzielnych plikach i są automatycznie kompilowane w momencie uruchamiania głównego programu. Biblioteka VTK umożliwia przypisanie obiektowi programów cieniujących (ang. *shader*) napisanych w języku Cg lub GLSL (Kitware Inc., 2011). Programy te mają ogromne możliwości wpływu na wygląd tworzonego obrazu (Dempski i Viale, 2005). Umożliwiają między innymi: uwzględnienie tekstur przy obliczeniach np. rozbłyśków (mapa połysku), nakładanie i łączenie wielu tekstur lub też zmianę geometrii w zależności od wartości tekstury.

Wykorzystanie programów cieniujących (ang. *shader*) jest powszechne w grach komputerowych oraz programach 3D, np. w programach CAD wspomagających projektowanie. W opracowanym fantomie drzewa oskrzelowego programy cieniujące służą do obliczania koloru ścian drzewa oskrzelowego (ang. *pixel shader*), uwzględniającego przyjęty model cieniowania, tekstury i model oświetlenia, a także umożliwiają zmianę współrzędnych powierzchni drzewa oskrzelowego (ang. *vertex shader*).

### 3.2.3.1 Model cieniowania

Otrzymanie realistycznie wyglądającej wirtualnej powierzchni wymaga uwzględnienia interakcji oświetlenia z tą powierzchnią — padające promieniowanie może być odbite, rozproszone, pochłonięte lub załamane (np. w ośrodku przezroczystym). W grafice komputerowej można spotkać bardzo wiele różnych modeli cieniowania, które umożliwiają generowanie zarówno



obrazów bardzo realistycznych, jak również zupełnie sztucznych (np. cieniowanie naśladujące rysunek odręczny) (Randima i Kilgard, 2003; Dempki i Viale, 2005). To właśnie one odpowiadają za wyznaczanie kolorów poszczególnych fragmentów obiektów trójwymiarowej sceny, a zatem również za wygląd generowanego obrazu wirtualnej sceny.

**Eksperymentalny model cieniowania BRDF.** Do opisu sposobu interakcji oświetlenia z powierzchnią odbijającą światło stosuje się *dwukierunkową funkcję rozkładu odbicia*, nazywaną funkcją BRDF (ang. *Bidirectional Reflectance Distribution Function*) (Nicodemus, 1965; Dempki i Viale, 2005). Opisuje ona stosunek luminancji energetycznej mierzonej w kierunku obserwatora  $dL(\vec{V})$  do natężenia luminancji padającej z kierunku oświetlenia  $dE(\vec{L})$  (rysunek 3.8):

$$f_{BRDF}(\vec{L}, \vec{V}) = \frac{dL(\vec{V})}{dE(\vec{L})}. \quad (3.26)$$

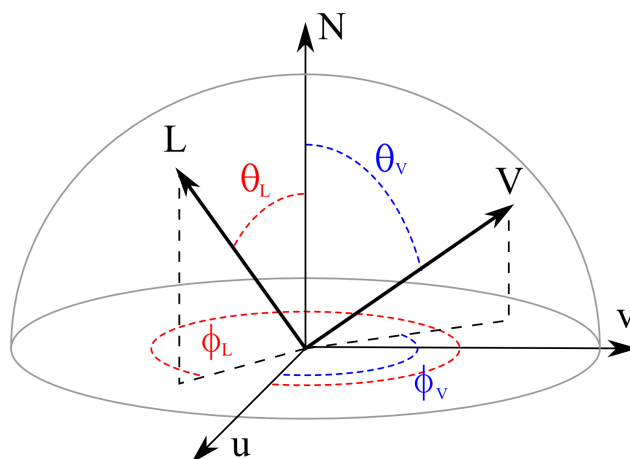
Funkcje BRDF można podzielić na dwie grupy: opracowane z uwzględnieniem zjawisk fizycznych oraz opisujące pomiary eksperymentalne. Pierwsza grupa skupia się na modelowaniu teorii fizycznego oddziaływania światła z opisywaną powierzchnią, np. uwzględnia matematyczny opis chropowatości powierzchni. Natomiast drugą grupę stanowią modele eksperymentalne, których struktura i współczynniki zostały dobrane tak, by „pasowały” do danych pomiarowych oraz by dawały „dobry efekt”. Zazwyczaj nie mają one uzasadnienia teoretycznego, mogą jednak stanowić dobrą aproksymację występujących w rzeczywistości zjawisk.

Dwukierunkowość funkcji BRDF oznacza, że zamiana obserwatora oraz źródła światła nie powodują zmiany opisu zjawiska. Warunek ten zachodzi, gdy  $f(\vec{L}, \vec{V}) = f(\vec{V}, \vec{L})$ . Przy modelowaniu rzeczywistych zjawisk optycznych funkcją BRDF, z uwzględnieniem zasad fizyki, niezbędne jest uwzględnienie zasady zachowania energii, zgodnie z którą suma całkowitej energii wypromieniowanej na skutek odbicia światła od powierzchni nie może być większa niż energia światła padającego. Nie wszystkie modele uwzględniają tę zasadę.

Współrzędne wektorów w równaniu (3.26) określane są zazwyczaj w sferycznym układzie współrzędnych, przez podanie dwóch kątów  $\theta$  liczonych względem wektora normalnego powierzchni oraz dwóch kątów  $\phi$  określających kierunek w lokalnym układzie współrzędnych  $(u, v)$  (rysunek 3.8). W takiej postaci funkcja BRDF jest więc funkcją czterowymiarową. Uwzględniając różnorodność właściwości optycznych modelowanego materiału w różnych jego punktach, BRDF staje się funkcją sześciu zmiennych. Do opisu powierzchni transparentnych stosuje się dwukierunkową funkcję rozkładu przepuszczalności światła (ang. *Bidirectional Transmittance Distribution Function* — BTDF).

Wartość funkcji BRDF w pełni charakteryzuje własności refleksyjne powierzchni odbijającej promieniowanie i może być wyznaczona eksperymentalnie dla szerokiego spektrum promieniowania lub też dla wybranych długości odpowiadających np. składowym RGB. Uwzględnienie w funkcji BRDF, opisującej konkretną powierzchnię, długości fali promieniowania padającego oraz odbitego umożliwia opis takich zjawisk jak luminescencja oraz iryzacja powierzchni. Możliwe jest również dodanie kolejnych wymiarów funkcji, takich jak np. dwuwymiarowe współrzędne punktu na powierzchni, dla którego wyznaczana jest funkcja, co umożliwia uwzględnienie w modelu niejednorodności występujących na powierzchni. Eksperymentalne wyznaczenie wartości funkcji BRDF nie jest zadaniem łatwym, szczególnie w przypadku powierzchni „organicznych”, które charakteryzują się występowaniem bardzo skomplikowanych zjawisk optycznych, np. *rozproszeniem podpowierzchniowym* (ang. *subsurface scattering*). Polega ono na wielokrotnym rozpraszaniu światła na wielu warstwach o różnych właściwościach optycznych. Na skutek wielokrotnych odbić promień światła może się „wydo-





Rysunek 3.8: Dwukierunkowa funkcja rozkładu odbicia (BRDF) jako funkcja kątów padania światła ( $\theta_L, \phi_L$ ) oraz kątów obserwacji ( $\theta_V, \phi_V$ )

stać” w innym miejscu. Zjawisko takie może być modelowane ośmiowymiarową wersją funkcji BRDF o nazwie *Bidirectional Surface Scattering Reflectance Distribution Function* — *BSSRDF*, uwzględniającą wektor padania światła i jego współrzędne na powierzchni oraz wektor obserwacji wraz ze współrzędnymi wyjścia światła. W pracy (Donner i Jensen, 2006) przedstawiono przykład użycia funkcji BSSRDF do modelowania wyglądu powierzchni skóry człowieka. Zaproponowany model uwzględnia zjawisko rozproszenia powierzchniowego oraz podpowierzchniowego, a jego parametrami są proste w interpretacji wielkości: oleistość powierzchni skóry, stężenie hemoglobiny i melaniny oraz współczynnik uwzględniający występowanie różnych typów melaniny.

**Model cieniowana Cooka i Torrance’a.** Zaproponowany przez Cooka i Torrance’a w 1981 roku model oświetlenia jest modelem opracowanym w oparciu o fizykę odbicia (Cook i Torrance, 1981). Autorzy modelują chropowatość powierzchni w postaci mikrowgłębieni w kształcie litery V, bazując na modelu powierzchni Torrance’a-Sparrowa (Torrance i Sparrow, 1967). W modelu tym zakłada się, że powierzchnia zbudowana jest z ogromnej ilości rozmieszczonych izotropowo mikrowgłębieni, ścianki mikrowgłębieni są nachylone pod tym samym kątem oraz idealnie odbijają światło. Rozmieszczenie mikrościanek określa *funkcja szorstkości*  $D$ . W najprostszej wersji jest to funkcja Gaussa, a w modelu Cooka-Torrance’a jest rozkładem Beckmanna. Model uwzględnia dwa odbicia: rozproszone, opracowane na bazie modelu Lamberta oraz zwierciadlane, które uwzględnia takie właściwości materiału, jak jego szorstkość, geometryczne wygaszanie odbicia oraz współczynnik Fresnela.

Wprowadzenie mikrowgłębieni umożliwia uwzględnienie w modelu oświetlenia odbicia bezpośredniego od powierzchni, jak również odbicie niebezpośrednie (wielokrotne) występujące między mikrościankami. Dodatkowo, geometria mikrościanek może blokować światło poprzez zjawisko cieniowania i maskowania światła, powodując osłabienie odbicia zwierciadlanego. Efekt ten modeluje *współczynnik geometrycznego wygaszania*  $G$ . Kolejnym elementem uwzględnionym w tym modelu jest współczynnik Fresnela, który określa stosunek energii światła odbitego do energii światła padającego. Współczynnik ten zależy od kąta padania i długości fali. Umożliwia on modelowanie rozbłysków światła pojawiających się na powierzchniach obserwowanych pod niewielkim kątem.

Ogólne równanie opisujące model oświetlenia Cooka-Torrance'a ma postać:

$$f(\vec{L}, \vec{V}) = \frac{1}{\pi} \frac{F \cdot G \cdot D}{(\vec{N} \circ \vec{L}) \cdot (\vec{N} \circ \vec{L})}, \quad (3.27)$$

gdzie:  $F$  to współczynnik Fresnela,  $G$  opisuje tłumienie geometryczne, zaś  $D$  jest funkcją szorstkości, rozmieszczenia mikrowgłębień (rozkład Beckmanna).

Współczynnik Fresnela jest zazwyczaj aproksymowany funkcją zaproponowaną przez Schlicka (Schlick, 1994) w postaci:

$$F(\Theta) = F_0 + (1 - F_0) \cdot (1 - \cos(\Theta))^5, \quad (3.28)$$

gdzie  $F_0$  jest wartością współczynnika dla zerowego kąta  $\Theta$  i określonej długości fali, odczytaną z tablic materiałowych.

Z kolei tłumienie geometryczne  $G$  wyznaczane jest z następującej zależności:

$$G = \min \left\{ 1, \frac{2 \cdot (\vec{N} \circ \vec{H}) \cdot (\vec{N} \circ \vec{V})}{(\vec{V} \circ \vec{H})}, \frac{2 \cdot (\vec{N} \circ \vec{H}) \cdot (\vec{N} \circ \vec{L})}{(\vec{V} \circ \vec{H})} \right\}, \quad (3.29)$$

gdzie:  $\vec{N}$  wektor normalny do powierzchni,  $\vec{L}$  wektor w kierunku światła,  $\vec{V}$  wektor w kierunku obserwatora,  $\vec{H}$  wektor połowkowy między  $\vec{L}$  i  $\vec{V}$ .

Funkcja rozkładu mikropowierzchni  $D$  zastosowana w modelu ma postać:

$$D = \frac{e^{-\left(\frac{tg^2(\beta)}{m_B^2}\right)}}{4 \cdot m_B^2 \cdot \cos^4(\beta)}, \quad (3.30)$$

gdzie  $m_B \in (0, 1)$  opisuje właściwości powierzchni. Mała wartość odpowiada powierzchni gładkiej.

Kombinacja współczynników modelu umożliwia sterowanie rozbłyskiem w szerszym zakresie niż ma to miejsce w przypadku modelu Phong'a (strona 41). Model ten posiada cechy ogólnej funkcji BRDF, w tym spełnia zasadę wzajemności oraz zachowania energii. Podstawową wadą tego modelu są duże koszty obliczeniowe.

**Model cieniowana Orena i Nayara.** Model Orena i Nayara, zaproponowany w 1994 roku (Oren i Nayar, 1994), ma podobne założenia jak model Cooka i Torrance'a. Również w nim powierzchnia modelowana jest w postaci mikrowgłębień, z tą różnicą, że zamiast lustrzanego odbicia modeluje się je jako powierzchnie rozpraszające światło w sposób lambertowski. W tym celu autorzy zastosowali rozkład Gaussa kierunku wektora normalnego. W odróżnieniu od modelu Cooka-Torrance'a, model ten skupia się wyłącznie na zjawisku rozproszenia światła, bez modelowania rozbłysków. Znalazł on głównie zastosowanie w modelowaniu matowych powierzchni.

Równanie opisujące model Oren-Nayar'a ma następującą postać:

$$f(\vec{L}, \vec{V}) = \frac{k_d}{\pi} (A + B \cdot \max\{0, \cos(\phi_L - \phi_V)\} \cdot \sin(\Theta_{max}) \cdot \sin(\Theta_{min})), \quad (3.31)$$

gdzie :

$$A = 1 - \frac{\delta^2}{2 \cdot (\delta^2 + 0,33)}, \quad (3.32)$$

$$B = \frac{0,45 \cdot \delta^2}{\delta^2 + 0,09}, \quad (3.33)$$

$$\Theta_{max} = \max\{\Theta_L, \Theta_V\}, \quad (3.34)$$

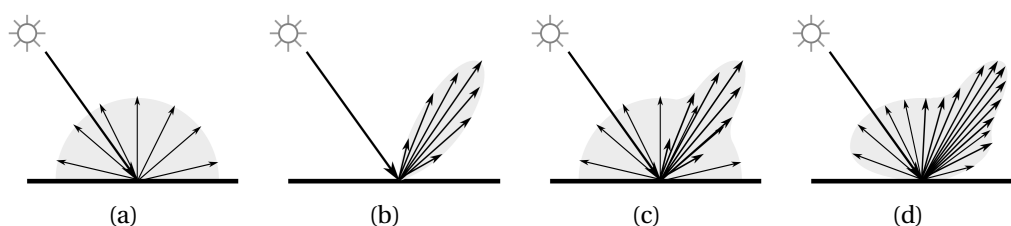
$$\Theta_{min} = \min\{\Theta_L, \Theta_V\}, \quad (3.35)$$

zaś  $k_d$  jest współczynnikiem odbicia rozproszonego, a  $\delta$  jest stałą materiałową, która modeluje matowość powierzchni (rozproszenie światła od mikrowgłębień).

**Empiryczne modele cieniowania.** Przytoczone powyżej fizykalne modele cieniowania dają duże możliwości modelowania powierzchni zarówno lustrzanych, jak i matowych. Istnieją jednak powierzchnie, które wymagają jeszcze bardziej skomplikowanego opisu. Można do nich zaliczyć powierzchnie wielowarstwowe, takie jak skóra lub inne tkanki (Donner i Jensen, 2006). Uwzględnienie tak skomplikowanych zjawisk w modelu powierzchni drzewa oskrzelowego wydaje się nieuzasadnione, zwłaszcza biorąc pod uwagę znaczne koszty obliczeniowe oraz trudności w wyznaczeniu parametrów dla tego typu zjawisk. Z przeprowadzonych testów wynika, że czas generowania obrazu ścian drzewa oskrzelowego z uwzględnieniem rozproszenia podpowierzchniowego wynoszący kilka sekund jest zbyt długi, by taki model oświetlenia mógł być użyty do testowania algorytmów estymacji ruchu.

Możliwości opisu powierzchni przy użyciu różnych wersji funkcji BRDF są bardzo duże i są wykorzystywane między innymi w generowaniu fotorealistycznych obrazów przeznaczonych do przemysłu filmowego. Ze względu na bardzo dużą złożoność obliczeniową, zastosowanie funkcji rozkładu odbicia do wydajnego generowania obrazów w czasie rzeczywistym jest bardzo utrudnione. Dlatego też do wydajnego tworzenia obrazów w czasie rzeczywistym używa się zazwyczaj prostszych, empirycznych modeli cieniowania, które mimo znacznych uproszczeń dają zadowalające rezultaty.

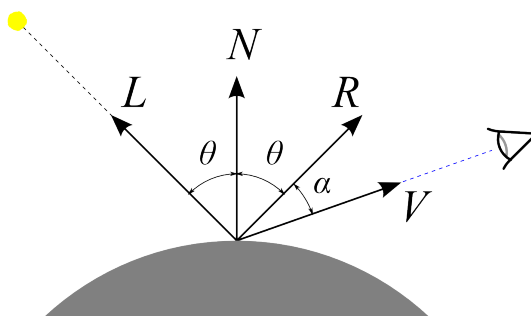
**Model cieniowania Phong.** Szczególne znaczenie dla szybkiego tworzenia grafiki komputerowej ma empiryczny model oświetlenia opracowany w 1975 roku przez *Phong Bui-Tuonga*. Model ten, nazwany od nazwiska twórcy *modelem oświetlenia Phong*, dość dobrze opisuje zjawiska zwierciadlanego odbicia światła od nieidealnie gładkich powierzchni. Zakłada on, że kolor powierzchni jest wynikiem: *odbicia rozproszonego*, które zabarwia padające światło na kolor obiektu oraz *odbicia zwierciadlanego* padającego światła (bez zmiany koloru). Graficzną interpretację tych zjawisk przedstawiono na rysunku 3.9. Model ten odpowiada np. powierzchniom plastikowym. W modelach oświetlenia uwzględnia się również współczynnik od-



Rysunek 3.9: Schemat powstawania odbicia od nieidealnie równej powierzchni: (a) model odbicia rozproszonego (ang. *diffuse*), (b) model odbicia zwierciadlanego (ang. *specular*), (c) suma modeli odbicia rozproszonego i zwierciadlanego oraz (d) przykład rzeczywistego odbicia

bicia światła otaczającego, które dociera do obiektu z każdej strony. Kolor obiektu w modelu oświetlenia Phong jest więc sumą współczynników oświetlenia: otoczenia, rozproszenia oraz rozbłysku. Każdy z tych współczynników jest zależny od kombinacji właściwości materiałowych oraz właściwości światła (Dempski i Viale, 2005).

Na rysunku 3.10 przedstawiono podstawowe wektory używane do wyznaczenia koloru powierzchni z użyciem modelu oświetlenia Phong. Wektor normalny  $N$  wyznacza kierunek pro-



Rysunek 3.10: Wektory używane do wyznaczenia współczynników modelu oświetlenia Phong:  $N$  - wektor normalny,  $L$  - wektor określający kierunek padania światła,  $R$  - promień odbity,  $V$  - kierunek do obserwatora

stopadły do płaszczyzny stycznej do powierzchni w danym punkcie. Wektor normalny może być wyznaczany analitycznie, np. na podstawie współrzędnych sąsiednich wierzchołków, może być również przypisany do wierzchołka siatki opisującej powierzchnię jako parametr (trójwymiarowy unormowany wektor). Możliwe jest również zakodowanie wektora normalnego w postaci tekstury mapowanej na powierzchnię. Wektor  $L$  wyznacza kierunek od rozpatrywanego punktu powierzchni, dla którego obliczany jest kolor do źródła światła. Wektor  $R$  jest odbitym promieniem światła  $L$ , zaś wektor  $V$  wyznacza kierunek od analizowanego punktu obiektu do obserwatora, symbolicznie zaznaczonego okiem.

Równanie modelu Phong'a (3.36) opisuje luminancję konkretnego punktu obiektu w funkcji długości fali  $L_v(\lambda)$  (Zabrodzki, 1994):

$$L_v(\lambda) = \underbrace{K_a(\lambda)L_a(\lambda)}_{\text{światło otaczające}} + \underbrace{K_d(\lambda)(\vec{N} \cdot \vec{L})L(\lambda)}_{\text{światło rozproszone}} + \underbrace{K_s(\vec{R} \cdot \vec{V})^{N_s}L(\lambda)}_{\text{światło odbite zwierciadlanie}}, \quad (3.36)$$

gdzie:  $K_a$  jest współczynnikiem odbicia światła otoczenia (ang. *ambient*),  $L_a$  luminancją światła otoczenia,  $K_d$  to współczynnik rozproszonego odbicia światła,  $L$  luminancja źródła światła,  $K_s$  współczynnik zwierciadlanego odbicia światła, zaś  $N_s$  określa współczynnik gładkości powierzchni.

W równaniu (3.36) można wyróżnić trzy człony. Pierwszy odpowiada za odbicie światła otaczającego  $L_a$ , docierającego do obiektu ze wszystkich stron. Światło to modeluje bezkierunkowe oświetlenie tła o stałym, zazwyczaj niewielkim natężeniu. Drugi człon, zawierający współczynnik rozproszenia światła  $K_d$ , modeluje mikroskopową chropowatość powierzchni, na której padające światło odbija się we wszystkich kierunkach w równym stopniu. Natężenie odbitego światła jest proporcjonalne do właściwości materiału (współczynnika  $K_d$ ), kąta padania światła na powierzchnię (iloczynu skalarnego znormalizowanych wektorów  $N$  i  $L$ ) oraz luminancji źródła światła. Człon ten modeluje rozproszenie światła z zachowaniem prawa Lamberta, zgodnie z którym światłość promieniowania odbitego jest proporcjonalna do kosinusa kąta padania. Rzeczywiste powierzchnie rozpraszające zachowują się zgodnie z tym prawem jedynie w pewnym zakresie kątów. Mimo tej wady, taki opis odbicia rozproszonego jest stosowany najczęściej. Ostatni człon odpowiada za modelowanie rozbłysków i zależy głównie od pozycji obserwatora względem obiektu i światła oraz parametrów powierzchni  $K_s$  oraz  $N_s$ . Współczynnik gładkości powierzchni umożliwia regulację wielkości rozbłysku — obiekty gładkie mają bardzo mały rozbłysk (duże  $N_s$ ). Równanie (3.36) nie uwzględnia wielu źródeł światła oraz przewiduje jedynie projekcję równoległą, a nie perspektywiczną.

Rozszerzenie modelu oświetlenia Phong'a o kolorowe rozbłyski  $K_s(\lambda)$  i projekcję perspektywiczną ma następującą postać:

$$L_v(\lambda) = K_a(\lambda)L_a(\lambda) + K_d(\lambda) \sum_{n=1}^{l_s} (\vec{N} \cdot \vec{L}_n)L_n(\lambda) + K_s(\lambda) \sum_{n=1}^{l_s} (\vec{N} \cdot \vec{H}_n)^{N_s}L_n(\lambda). \quad (3.37)$$

Występujący we wzorze (3.37) wektor  $H$  leży na dwusiecznej kąta tworzonego przez wektor światła  $L$  i wektor obserwatora  $V$  (rysunek 3.10). Odpowiada on wektorowi normalnemu płaszczyzny idealnego odbicia. Iloczyn skalarnych wektorów  $N$  oraz  $H$  uwzględnia projekcję perspektywiczną. Dla wszystkich światel występujących w scenie ( $l_s$ ) wyznaczane są składowe odpowiedzialne za światło rozproszone oraz światło odbite. Model ten może być również uzupełniony o efekt samoświecenia powierzchni, który realizowany jest przez dodanie do równania (3.37) stałej wartości.

Równanie (3.37) stosuje się zazwyczaj do obliczenia trzech składowych koloru: R, G i B. Również parametry materiałowe  $K_a$ ,  $K_d$ ,  $K_s$  określane są jako trójelementowe wektory, zawierające współczynniki dla składowych modelu RGB.

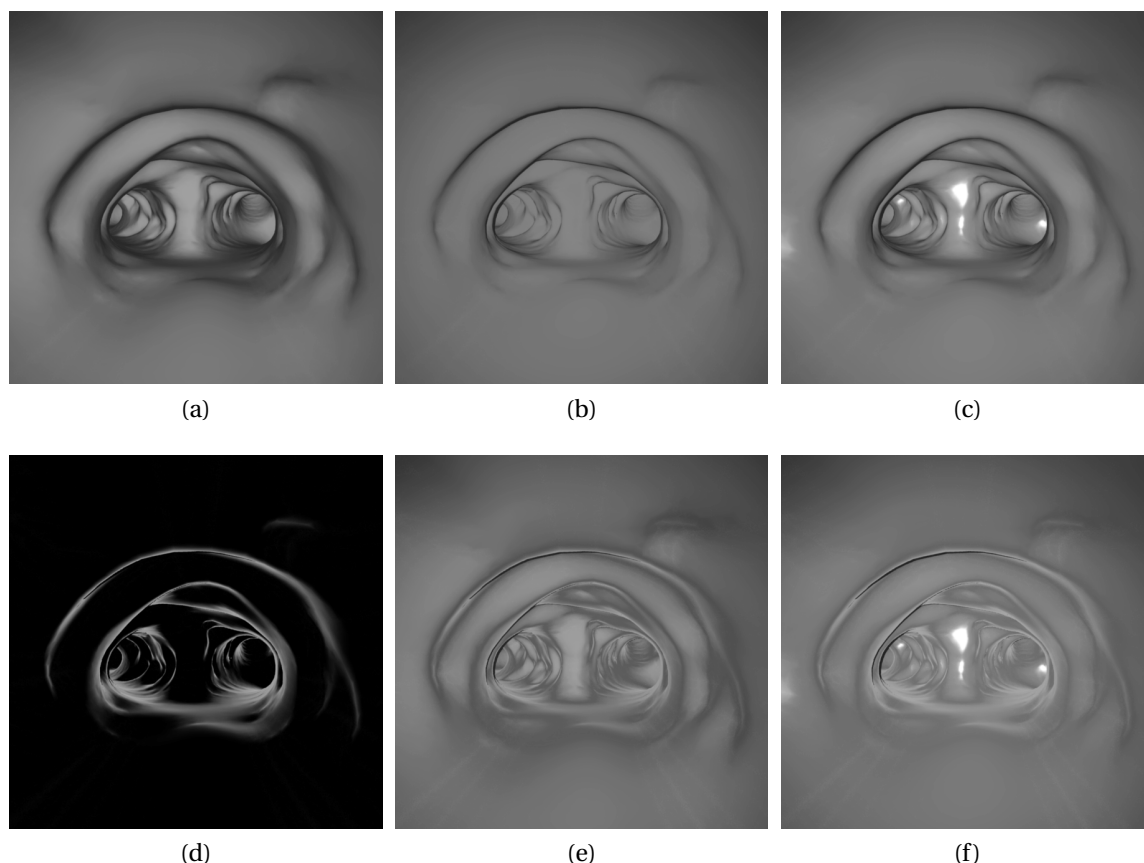
Model oświetlenia Phong'a bez dodatkowych ograniczeń na współczynniki nie spełnia zasady zachowania energii ani wzajemności (Lewis, 1994), a mimo to jest powszechnie stosowany w grafice komputerowej do wyliczania koloru oświetlonych obiektów 3D. Popularność swą zawdzięcza relatywnie prostemu opisowi matematycznemu, który mógł być realizowany w postaci sprzętowej w pierwszych akceleratorach grafiki komputerowej. Również dzisiaj, w dobie programowalnych procesorów graficznych, metoda cieniowania Phong jest nadal powszechnie stosowana, ponieważ jest wydajna i daje dobre rezultaty.

Podstawową wadą modelu oświetlenia Phong'a jest „plastikowość” uzyskiwanych obrazów. Model ten jest zbyt ubogi, by można go było używać do symulowania powierzchni materiałów o bardziej skomplikowanej budowie, takich jak tkaniny czy skóra. Na bazie modelu Phong'a powstało wiele innych modeli oświetlenia, które są wprawdzie bardziej kosztowne obliczeniowo, jednak uwzględniając więcej zjawisk optycznych umożliwiają generowanie fotorealistycznych obrazów.

Zmodyfikowaną wersją modelu Phong'a jest model Blinna-Phong'a (Blinn, 1977), w którym uzyskano większą wydajność poprzez wyeliminowanie obliczania wektora odbicia zwierciadlanego. Zamiast niego, Blinn zaproponował użycie w obliczeniach wektora połówkowego, znajdującego się pośrodku, między kierunkiem światła  $\vec{L}$  a obserwatorem  $\vec{V}$ .

**Porównanie modeli cieniowania.** W celu przetestowania różnych modeli oświetlenia dokonano subiektywnego porównania jakości obrazów uzyskanych dla tego samego modelu powierzchni drzewa oskrzelowego oraz różnych modeli oświetlenia. Rysunek 3.11 przedstawia obrazy uzyskane przy stałym oświetleniu, pozycji kamery oraz modelu powierzchni drzewa oskrzelowego. Współczynniki poszczególnych modeli były dobrane w taki sposób, by uwypuklić właściwości poszczególnych modeli cieniowania.

Można zauważyć, że modelowanie rozproszenia światła zgodnie z prawem Lamberta (rysunek 3.11a) generuje obraz o dużym kontraście, w którym ściany drzewa oskrzelowego widoczne pod ostrym kątem są zaciemnione. Uwzględnienie „szorstkości” materiału (współczynnik szorstkości równy 0,5, dostępny przedział od 0 do 1) w modelu Oren-Nayara powoduje zmniejszenie kontrastu obrazu (rysunek 3.11b). Obraz ścian drzewa oskrzelowego wygląda na matowy, zaś ciemne obszary ścian są mniejsze i wyglądają jakby były oświetlone światłem odbitym od sąsiednich ścian. Dodanie do modelu rozbłysków (rysunek 3.11c) powoduje „rozświetlenie” ścian skierowanych prostopadle do kamery oraz znajdującego się obok niej źródła



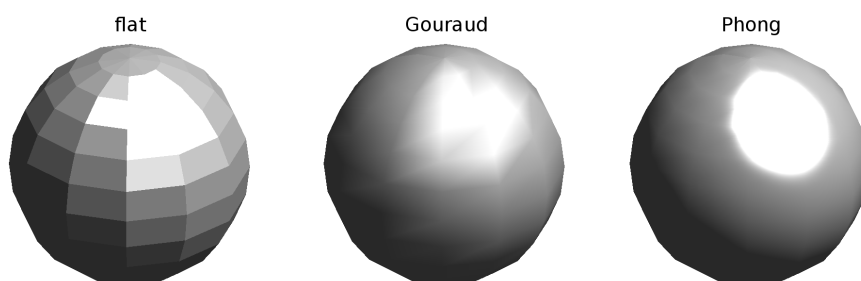
Rysunek 3.11: Porównanie modeli oświetlenia: (a) model rozproszenia lambertowskiego – cieniowanie Phong’a, (b) model Oren-Nayara, (c) model Oren-Nayara z dodanymi odbiciami zwierciadlanymi, (d) zjawisko Fresnela w modelu Cook-Torrance’a, (e) światło rozproszone oraz efekt Fresnela modelu Cook-Torrance’a, (f) pełen model Cook-Torrance’a obejmujący światło rozproszone, odbicie zwierciadlane oraz zjawisko Fresnela

światła. Rozbłyski sugerują, że światło odbiło się od „mokrej” ściany. Uwzględnienie zjawiska Fresnela w modelu oświetlenia powoduje „rozświetlenie” ścian zwróconych pod kątem ostrym do kamery (rysunek 3.11d). Efekt ten, w połączeniu z oświetleniem rozproszonym (rysunek 3.11e) oraz rozbłyskami (rysunek 3.11f) może posłużyć, do modelowania tkanek pokrytych przezroczystą wydzieliną. W takim podejściu, półprzezroczyste tkanki traktowane są jak matowa powierzchnia rozpraszająca światło, zaś śluz jako błyszcząca warstwa, która odbija światło padające zarówno prostopadle do powierzchni (odbicie zwierciadlane), jak również odbija światło padające na powierzchnię pod kątem ostrym (całkowite odbicie od powierzchni, efekt Fresnela).

Zastosowany przez autora rozprawy model oświetlenia jest zmodyfikowanym pod kątem wydajności oraz uwzględniającym teksturowanie modelem odbicia rozproszonego Oren-Nayar’a. Dodano do niego również odbicie zwierciadlane obliczane metodą Blinna-Phonga oraz efekt Fresnela aproksymowany funkcją Schlicka. Taki model oświetlenia uwzględnia większość występujących na powierzchni rzeczywistego drzewa oskrzelowego zjawisk, takich jak: rozproszenie światła w półprzezroczystych komórkach (aproksymowane szorstkością w modelu Oren-Nayar’a), pojawianie się rozbłysków światła na mokrej powierzchni tkanek pojawia-

jące się „na wprost” kamery (rozbłysk zwierciadlany) oraz rozbłyski na krawędziach (efekt Fresnela).

**Interpolacja koloru.** Poza modelem cieniowania, kluczowe znaczenie w generowaniu grafiki 3D ma również tzw. *cieniowanie* wielokątów, czyli wyliczanie koloru pikseli należących do konkretnych wielokątów opisujących powierzchnię. Ponieważ obliczenia koloru punktu powierzchni są kosztowne obliczeniowo (wyliczenie koloru z modelu oświetlenia), stosuje się interpolację koloru, która umożliwia przyspieszenie obliczeń. Na rysunku 3.12 przedstawiono wyniki użycia trzech metod cieniowania — interpolacji koloru powierzchni. We wszystkich przypadkach użyto tego samego modelu oświetlenia Phonga, jednak był on wyznaczany dla różnych obszarów. W metodzie *flat* kolor wynikający z modelu oświetlenia Phonga obliczany jest dla każdego wielokąta jednokrotnie, zazwyczaj dla środka wielokąta. Do obliczeń brany jest wektor normalny obliczony ze współrzędnych przestrzennych wierzchołków lub obliczany jako wektor wypadkowy wektorów przypisanych do wierzchołków.



Rysunek 3.12: Trzy rodzaje interpolacji (cieniowania) zastosowane podczas obliczania koloru powierzchni sfery

Druga metoda interpolacji koloru, zwana metodą *cieniowania Gourauda*, oblicza kolor modelu oświetlenia dla każdego wierzchołka wielokąta powierzchni. Ze względu na zasadę działania, metoda ta jest również nazywana metodą *per-vertex*. Następnie stosowana jest liniowa interpolacja koloru dla wszystkich pikseli wielokąta. Metoda ta umożliwia uzyskanie gładkich powierzchni jedynie przy bardzo gęstej siatce opisującej powierzchnię. Do poprawnej wizualizacji rozbłysków niezbędne są wierzchołki, ponieważ tylko „na nich” obliczany jest kolor wynikający z modelu oświetlenia. Na wygenerowanym obrazie widoczne są artefakty, które są wynikiem liniowej interpolacji koloru. Metoda cieniowania Gourauda, ze względu na małe koszty obliczeniowe oraz relatywnie dobrą jakość generowanego obrazu, była implementowana sprzętowo w pierwszych akceleratorach 3D.

Modela interpolacji (cieniowania) Phonga wylicza kolor każdego piksela wielokąta (ang. *per-pixel*) na podstawie interpolacji wektora normalnego, który z kolei wyliczany jest na podstawie wektorów normalnych wierzchołków. Metoda ta daje obraz najlepszej jakości (rysunek 3.12), zwłaszcza w obszarach, gdzie występują rozbłyski. Wysoka jakość jest jednak kosztowna obliczeniowo — kolor każdego piksela musi być wyliczony osobno, bezpośrednio z przyjętego modelu oświetlenia.

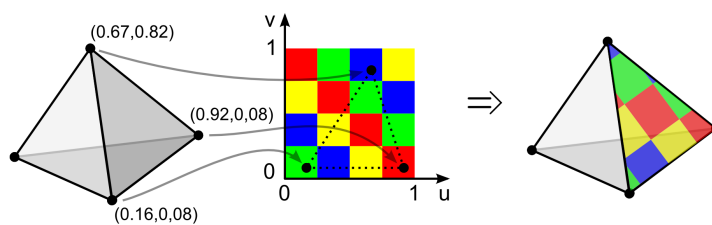
Ponieważ model powierzchni drzewa oskrzelowego zbudowany jest z dużej ilości trójkątów, które są niezbędne do odtworzenia szczegółów morfologicznych, zasadne wydaje się użycie szybkiego modelu cieniowania z interpolacją kolorów Gourauda. Jednak z przeprowadzonych testów wynika, że przyjęty szerokokątny model kamery powoduje, że w polu widzenia kamery pojawiają się powiększone trójkąty, które są niepoprawnie oświetlone. Dodatkowo, przyjęty

model oświetlenia uwzględniający zjawisko Fresnela, wymusza obliczenia modelu oświetlenia dla wszystkich pikseli (ang. *per-pixel*).

### 3.2.3.2 Mapowanie tekstur

Teksturowanie, nazywane również *nakładaniem tekstur*, jest techniką używaną w grafice komputerowej, której głównym celem jest zwiększenie szczegółów generowanego obrazu. W uproszczeniu można stwierdzić, że teksturowanie polega na „oklejaniu przestrzennego obiektu kolorową tapetą”. Teksturowanie polega na mapowaniu współrzędnych tekstury na współrzędne obiektu, na który tekstura ma być „nałożona”. Teksturą może być obraz bitmapowy w postaci mapy pikseli lub też może mieć ona postać funkcji matematycznej. Wówczas taką teksturę nazywa się teksturą proceduralną.

Rysunek 3.13 ilustruje zasadę teksturowania — mapowania współrzędnych tekstury (kolorowej szachownicy) na współrzędne trójwymiarowego obiektu (czworościanu). Wybranym wierzchołkom opisującym obiekt, poza trójwymiarowymi współrzędnymi w przestrzeni sceny ( $x, y, z$ ), przypisane są dodatkowe, dwuwymiarowe współrzędne ( $u, v$ ), wyznaczające punkty w przestrzeni tekstury. Dzięki temu poprzez interpolację, możliwe jest wyznaczenie koloru poszczególnych pikseli obiektu z uwzględnieniem koloru tekstury. Przypisane wierzchołkom



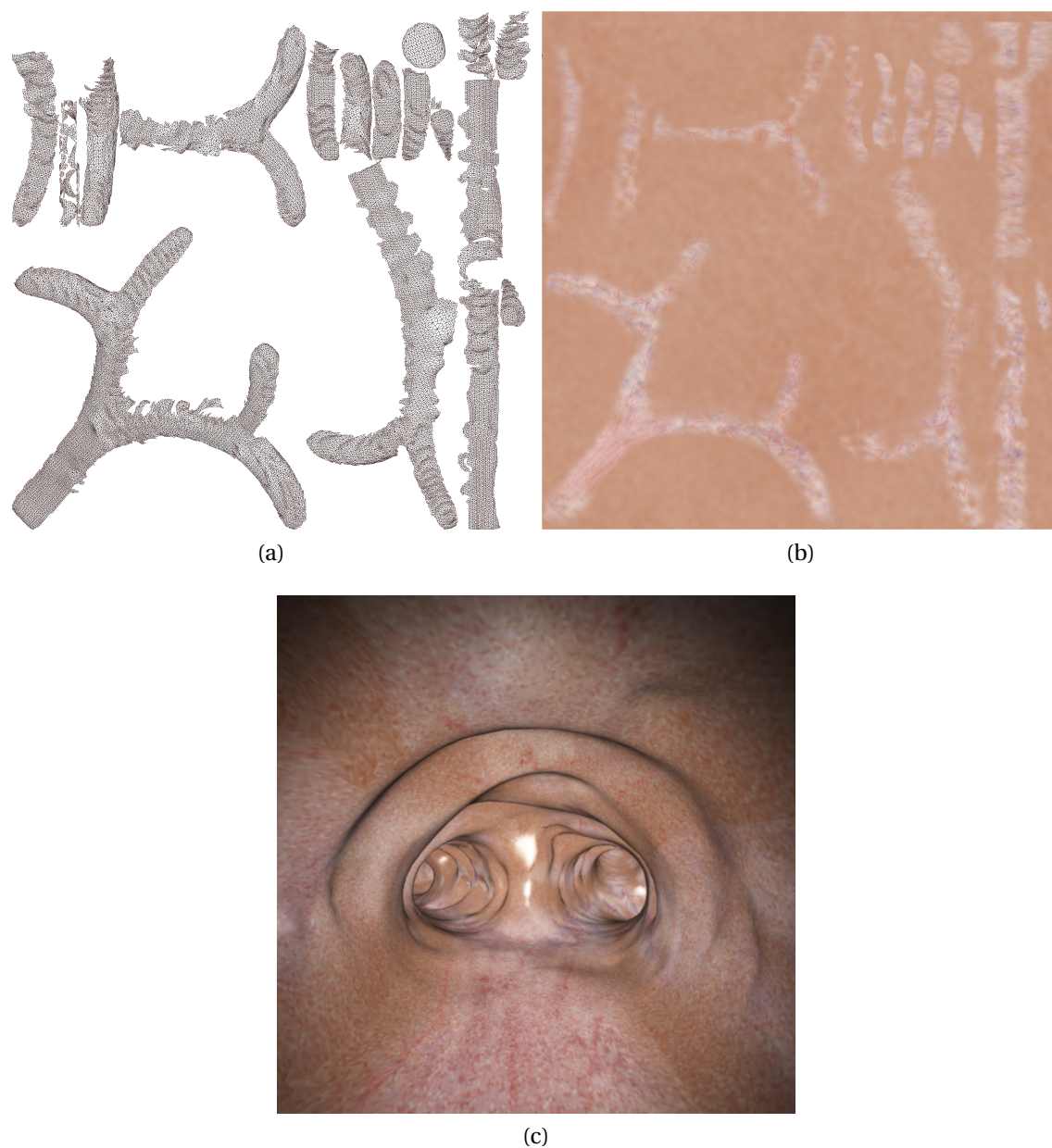
Rysunek 3.13: Zasada działania teksturowania przez mapowanie UV

współrzędne tekstury, zazwyczaj w postaci pary współrzędnych ( $u, v$ ) dla tekstur dwuwymiarowych lub ( $u, v, w$ ) dla tekstur trójwymiarowych, mogą być wyznaczane w różny sposób. Można rozróżnić dwie główne metody: współrzędne są przypisywane ręcznie, np. w trakcie modelowania przestrzennego obiektu autor modelu „przypina” teksturę do bryły. Wiąże się to zazwyczaj z „rozwinieniem” (ang. *unfold*) powierzchni obiektu na płaszczyznę, która służy do wyliczenia współrzędnych tekstury. W tej metodzie zazwyczaj nie jest wykorzystywana cała przestrzeń tekstury — występują obszary, które nie są mapowane.

Rysunek 3.14 przedstawia przykładową teksturę, która powstała przez rozwinięcie siatki wirtualnego drzewa oskrzelowego. Tekstura ta, zwana *UV mapą*, została wygenerowana w programie Blender dla siatki fantomu drzewa oskrzelowego.

Model powierzchni drzewa oskrzelowego, uwzględniający wiele szczegółów, musi być zbudowany z dużej liczby trójkątów. To z kolei prowadzi do trudności w automatycznym generowaniu map UV. Jak można zauważyć na przytoczonym rysunku 3.14a, rozwinięcie powierzchni drzewa oskrzelowego stanowiło wyzwanie nawet dla dedykowanego oprogramowania, jakim jest program Blender, służący do tworzenia grafiki komputerowej. Ze względu na dużą złożoność powierzchni, program podzielił siatkę na kilkadziesiąt fragmentów. Połączenie ich jest jak najbardziej możliwe, jednak wymaga sporego doświadczenia. Poważnym ograniczeniem tej metody nakładania tekstur jest to, że wygenerowana w taki sposób mapa UV może być użyta jedynie z siatką, dla której powstała. Zmiana kształtu modelu drzewa oskrzelowego lub np. dyskretyzacja modelu z mniejszym krokiem, wymaga wygenerowania nowej mapy UV i związanej z nią nowej tekstury.



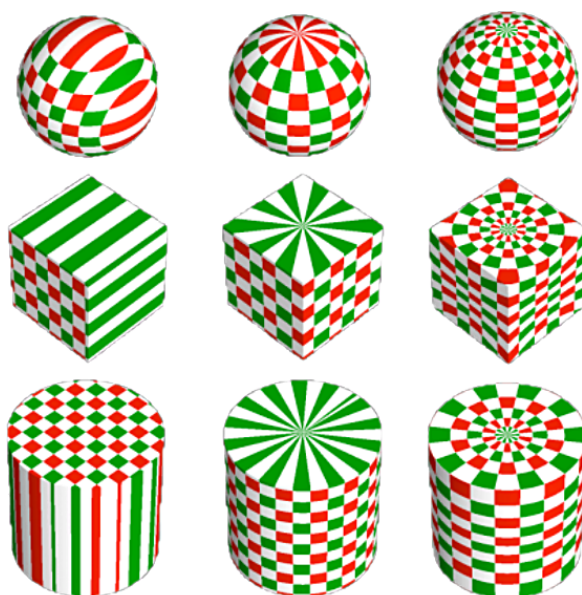


Rysunek 3.14: Przykład rozwinięcia siatki (a) wygenerowanej dla modelu drzewa oskrzelowego, (b) tekstura opracowana na jej podstawie oraz (c) obraz wirtualnego drzewa oskrzelowego z nałożoną teksturą

Na rysunku 3.14b przedstawiono przykład tekstury, która symuluje tkanki występujące w tchawicy. Została ona przygotowana w programie graficznym Gimp poprzez połączenie kilku warstw różnych fotografii skóry i innych tkanek. Dodatkowo dorysowano drobne linie, które symulują naczynia krwionośne. Do tak otrzymanej tekstury dodano szum, który symuluje mikroskopowe różnice kolorów, występujące w tkankach. Przykład użycia tak opracowanej tekstury zaprezentowano na rysunku 3.14c.

Drugą metodą generowania współrzędnych mapowania jest wyznaczenie współrzędnych tekstur w sposób analityczny, np. poprzez rzutowanie prostego obiektu (np. płaszczyzny) na teksturowany, bardziej skomplikowany obiekt. Jako obiekty pośrednie stosuje się takie, które mają prosty opis matematyczny, ułatwiający powiązanie współrzędnych tekstur z parametrycznym opisem obiektu pośredniego. Do najczęściej stosowanych zalicza się: płaszczyznę (tekstura opisana jest w prostokątnym układzie współrzędnych), walec (tekstura opisana jest w cylindrycznym układzie współrzędnych) oraz sferę (tekstura w sferycznym układzie współrzędnych). Mapowanie takie również nazywane jest *mapowaniem parametrycznym*. W zależności od użytego obiektu pośredniego, spotyka się również takie nazwy jak mapowanie sferyczne lub mapowanie cylindryczne.

Rysunek 3.15 przedstawia przykłady użycia mapowania tekstury przez rzutowanie płaszczyzny, walca oraz sfery na trzy bryły: sferę, sześcian oraz walec. Można zauważyć, że każde z zastosowanych mapowań daje inny efekt.

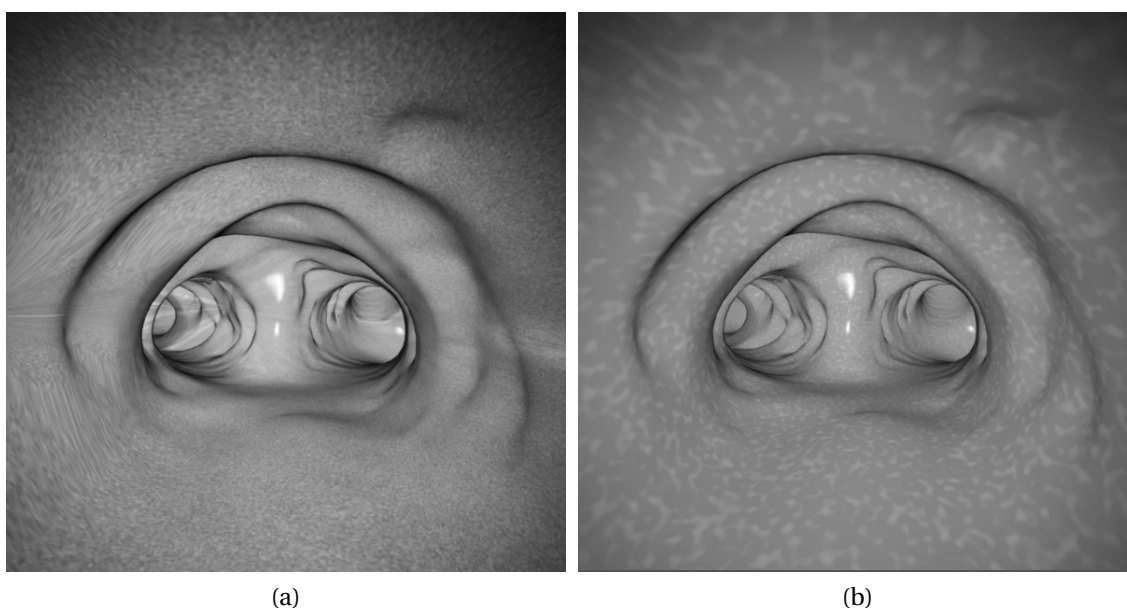


Rysunek 3.15: Przykłady mapowania tekstury na trzy rodzaje przestrzennych figur (sferę, sześcian oraz walec) metodą rzutowania płaszczyzny (pierwsza kolumna) walca (druga kolumna) oraz sfery (trzecia kolumna) (Bourke, March 1987)

Użycie mapowania parametrycznego, będącego wynikiem rzutowania prostej bryły pośredniej na bardziej skomplikowaną, może powodować artefakty objawiające się zniekształcaniem lub rozciąganiem tekstury. Występują one wówczas, gdy powierzchnie pomocnicza i obiektu są do siebie prostopadłe. Efekt taki można zauważyć na rysunku 3.15 na powierzchni bocznej walca przy mapowaniu płaszczyzną.

Na rysunku 3.16a przedstawiono przykład użycia tekstury z rysunku 3.14b, ale z pominięciem zapisanych współrzędnych UV, wykorzystując rzutowanie parametryczne w cylindrycz-

nym układzie współrzędnych. Na lewej ścianie tchawicy można zauważyć rozciągnięte fragmenty tekstury, które są artefaktami. Rysunek 3.16b przedstawia przykład użycia tekstury proceduralnej (szum Perlina), która generowana jest na bieżąco na podstawie opisującej ją funkcji matematycznej. Takie teksturowanie nie wymaga generowania współrzędnych mapowania, ponieważ zależą one zazwyczaj jedynie od współrzędnych w przestrzeni. Teksturowanie takie jest kosztowne obliczeniowo, a w przypadku zmiany geometrii obiektu (np. modelowanie ugięcia) może prowadzić do przesuwania się tekstury po obiekcie.



Rysunek 3.16: Przykład użycia: (a) mapowania parametrycznego oraz (b) tekstury proceduralnej (szum Perlina)

Pomimo występujących w mapowaniu parametrycznym (rysunek 3.16a) artefaktów może ono być z powodzeniem stosowane w fantomie drzewa oskrzelowego. Mapowanie parametryczne umożliwia w pełni automatyczne generowanie współrzędnych dla tekstur, niezależnie od stopnia skomplikowania topologii modelu oraz niezależnie od stopnia skomplikowania powierzchni. Takie podejście zostało zastosowane do przypisania tekstury w prostych modelach „tubowych”, stosowanych do testowania algorytmów estymacji położenia endoskopu.

### 3.3 Model kamery endoskopu

Zarówno ludzki wzrok, jak i układ do akwizycji obrazów (w skrócie *kamera*) dokonują pewnej transformacji rzeczywistej trójwymiarowej przestrzeni do własnej dwuwymiarowej przestrzeni obrazowania. Określenie parametrów tej transformacji jest niezbędne dla modelowania oraz analizy właściwości metrologicznych układu optycznego kamery. Matematyczne modele kamery można podzielić na dwie grupy: opierające się na fizycznych właściwościach i parametrach oraz modele nieparametryczne, bez bezpośredniej, fizycznej interpretacji ich współczynników. W prezentowanej pracy zastosowano podejście parametryczne, które umożliwiło powiązanie właściwości rzeczywistego, skomplikowanego źródła obrazu, jakim jest wideoendoskop z jego cyfrowym modelem.

Dla modeli fizycznych można wyróżnić dwie grupy parametrów opisujących kamerę:

1. *parametry zewnętrzne* — jednoznacznie określają przekształcenie układu współrzędnych kamery w układ współrzędnych zewnętrzny, związany zazwyczaj z obserwowaną sceną, można do nich zaliczyć:
  - a) wektor translacji, zaczepony w zewnętrznym (globalnym) układzie współrzędnych, określa pozycję kamery,
  - b) macierz rotacji, definiuje obroty lokalnego układu współrzędnych względem zewnętrznego układu współrzędnych, określa kierunek w jakim „patrzy“ kamera,
2. *parametry wewnętrzne*, w których można wyróżnić:
  - a) ogniskową  $f$  wyznaczającą własności rzutowania perspektywicznego dokonywanego przez kamerę,
  - b) sposób translacji pomiędzy współrzędnymi kamery a dyskretnymi współrzędnymi obrazu (pikselami), obejmuje położenie środka obrazu, liczbę i wymiary pikseli oraz ewentualne nierównoległe zamontowanie sensora obrazu w stosunku do płaszczyzny układu optycznego,
  - c) opis zniekształceń geometrycznych wprowadzanych przez układ optyczny kamery, którego źródłem są nieliniowości układu optycznego (abberacje sferyczne, koma, astygmatyzm, zniekształcenia poduszkowe i beczkowe pola widzenia), w większości przypadków można je zamodelować zniekształceniami radialnymi oraz tangencjalnymi.

Zewnętrzne parametry rzeczywistej kamery wideobronchofiberoskopu związane są z pozycją końcówki endoskopu. Wyznaczenie tych parametrów w trakcie badania, czyli obliczenie współrzędnych końcówki endoskopu względem np. łóżka, na którym leży pacjent, jest możliwe jedynie dla bezpośrednich systemów pomiarowych, np. elektromagnetycznych. Metody bazujące na obrazie pochodzącym z endoskopu umożliwiają jedynie określenie położenia końcówki względem otoczenia – w tym przypadku względem drzewa oskrzelowego pacjenta. W przypadku modelowania wirtualnej kamery endoskopu, parametry zewnętrzne kamery określane są w prosty sposób, przez podanie współrzędnych w globalnym układzie współrzędnych wirtualnej sceny. Macierz rotacji może być określana bezpośrednio przez podanie kątów obrotu lub pośrednio przez określenie w przestrzeni tzw. punktu skupienia, na który „patrzy” wirtualna kamera. W drugim przypadku, niezbędne jest również określenie tzw. wektora  $up$ , czyli kierunku jednoznacznie definiującego obrót kamery wokół osi prostopadłej do obrazu.

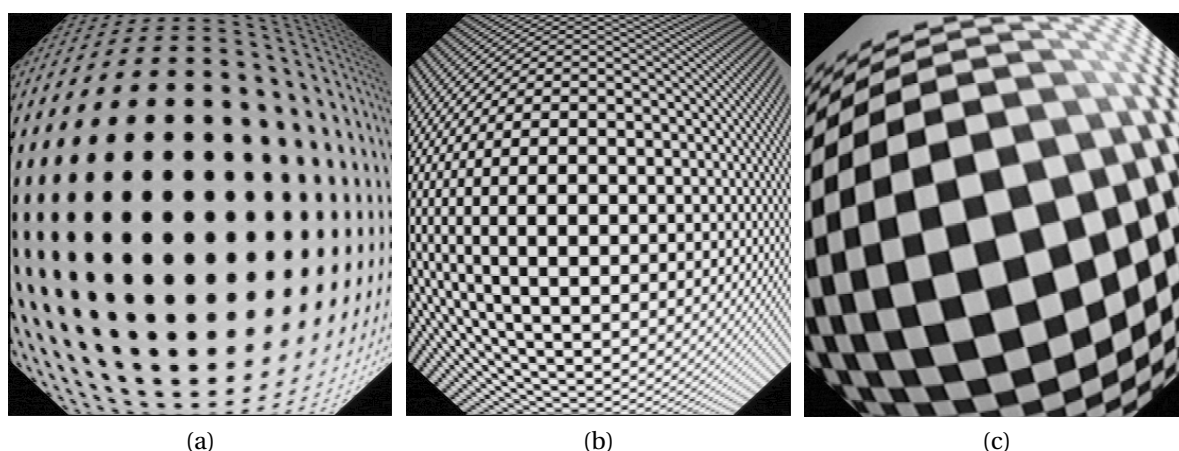
Parametry wewnętrzne rzeczywistego endoskopu zależą od jego budowy, zwłaszcza właściwości toru optycznego oraz sensora obrazu. Ogniskowa endoskopów jest zazwyczaj bardzo mała, co w połączeniu z niewielkimi rozmiarami sensorów optycznych daje bardzo szeroki kąt widzenia. Małą ogniskową układu optycznego osiąga się poprzez zastosowanie szerokokątnych soczewek, które z kolei wprowadzają bardzo duże zniekształcenia pola widzenia. Rozdzielczość obrazu determinowana jest zastosowanym w wideoendoskopie sensorem, zaś faktyczna rozdzielczość optyczna zależy od właściwości toru optycznego (soczewek) oraz liczby włókien zastosowanego światłowodu wizyjnego. Na podstawie przytoczonych faktów można stwierdzić, że model rzeczywistej kamery endoskopu musi uwzględniać wiele parametrów wewnętrznych, których wartości muszą być zidentyfikowane. Wirtualny model kamery endoskopu, stworzony do generowania obrazów wirtualnej endoskopii, bazuje na najprostszym modelu kamery – kamerze otworkowej, która zgodnie z teorią dokonuje jedynie przekształcenia perspektywicznego. Dlatego model ten musi być uzupełniony o występujące w rzeczywistej kamerze zjawiska, zwłaszcza takie jak nieliniowe zniekształcenia pola widzenia.

Identyfikacja parametrów wewnętrznych rzeczywistej kamery wideoendoskopu jest konieczna z dwóch powodów: do symulowania rzeczywistych zniekształceń na obrazach wirtualnych (wirtualny fantom) oraz w celu kompensacji istniejących zniekształceń. Przed analizą

ruchu własnego endoskopu niezbędne jest „wyprostowanie” rzutu ścian drzewa oskrzelowego widocznych na obrazach wideo. Dzięki temu, obraz może być analizowany w taki sposób, jakby był rejestrowany przez kamerę otworkową.

### 3.3.1 Dystorsja obrazu

Zapewnienie szerokiego kąta widzenia, umożliwiającego oglądanie ścian drzewa oskrzelowego przy pomocy endoskopu, wiąże się z koniecznością zastosowania szerokokątnego układu optycznego. Układy takie charakteryzują się znaczną dystorsją beczkową, która objawia się poprzez zakrzywianie linii prostych na rejestrowanym obrazie (tzw. efekt rybiego oka). Przykład obrazu płaskiej kartki zarejestrowanego przy użyciu endoskopu *Olympus BF 160* przedstawiono na rysunku 3.17. Endoskop skierowany został na płaską kartkę, na której wydrukowano regularne wzory: punkty oraz kratkę o różnej gęstości.



Rysunek 3.17: Przykład zniekształceń wprowadzanych przez tor optyczny wideobronchoskopu: (a) płaska powierzchnia z równomiernie rozmieszczonymi kopkami, (b) i (c) płaska powierzchnia z szachownicą o różnej gęstości

Jak można zauważyć, zniekształcenia geometrii wprowadzane do obrazu przez tor optyczny endoskopu są znaczne. Zniekształcenia te wymagają korekty w przypadku obrazu pochodzącego z endoskopu, na podstawie którego mają być wykonywane obliczenia pozycji endoskopu. Dodatkowo, w celu odwzorowania rzeczywistego charakteru obrazu bronchoskopowego na obrazach wirtualnej bronchoskopii, konieczne jest zasymulowanie zniekształceń tego typu na obrazach wirtualnych. Zarówno usuwanie, jak i nakładanie dystorsji geometrii obrazu wymaga wcześniejszej identyfikacji parametrów modelu opisującego te zniekształcenia.

### 3.3.2 Identyfikacja parametrów modelu zniekształceń

Zniekształcenia wprowadzane przez szerokokątną optykę kamery bronchoskopu modelowane są poprzez określenie środka zniekształceń oraz składowych radialnych i tangencjalnych (Shah i Aggarwal, 1996). Metodologia opracowanych metod bazuje na pracach (Asari i inni, 1999; Helferty i inni, 2001). Wykorzystano w niej dwie metody identyfikacji współczynników modelu

zniekształceń geometrii obrazu endoskopu (Socha i inni, 2005a). Pierwsza metoda wykorzystuje optymalizację średniokwadratową i uwzględnia jedynie zniekształcenia radialne. Natomiast druga metoda wykorzystuje bibliotekę ogólnego przeznaczenia (Bouguet, 2008), która umożliwia identyfikację rozszerzonego modelu kamery DLT (ang. *Direct Linear Transform*). Model ten uwzględnia nieliniowe zniekształcenia radialne i tangencjalne.

**Metoda dedykowana.** Na potrzeby korekcji zniekształceń obrazów pochodzących z endoskopu została opracowana metoda bazująca na metodologii przedstawionej w pracach (Asari i inni, 1999). Bazują one na analizie zarejestrowanych przez endoskop obrazów testowych, przedstawiających obraz płaskiej powierzchni, na której nadrukowano równomiernie rozmieszczone czarne kółka, których środki tworzą linie poziome i pionowe. Występowanie nieliniowych zniekształceń radialnych objawia się w postaci „zakrzywania” linii wyznaczanych przez czarne markery. Idea korekcji zniekształceń w tej metodzie polega na takim doborze środka zniekształceń oraz funkcji zmienności promienia, aby uzyskać maksymalnie proste linie (w sensie średniokwadratowym) na obrazach z endoskopu.

W przeciwieństwie do metodologii zaproponowanej w pracy (Asari i inni, 1999) środek zniekształceń wyznaczany jest poprzez wielokrotną optymalizację w okolicach środka geometrycznego obrazu (jest on punktem startowym algorytmu optymalizacji).

Metoda dedykowana dla endoskopów została zaimplementowana w programie Matlab i składa się z następujących kroków:

- wstępnego przetwarzania obrazu (filtracja dolnopasmowa oraz binaryzacja), którego celem jest wyodrębnienie środków markerów (średnia arytmetyczna współrzędnych pikseli markera),
- grupowania markerów w linie poziome i pionowe,
- ustalenia warunków początkowych algorytmu optymalizacji: przyjęcie modelu zniekształceń radialnych (wielomian stopnia czwartego) i środka zniekształceń (geometryczny środek obrazu),
- wyznaczenia współczynników modelu poprzez optymalizację średniokwadratową:
  - w kolejnych iteracjach do każdej linii poziomej i pionowej dopasowano prostą w sensie średniokwadratowym, a suma błędów tych dopasowań stanowiła wartość minimalizowaną; do minimalizacji użyto funkcji *fminunc()* programu Matlab,
  - optymalizację przeprowadzono ze względu na współczynniki wielomianu korekcyjnego czwartego rzędu:  $r_r(r) = \sum_{n=0}^4 a_n r^n$ .

Algorytm estymacji współczynników można podzielić na dwa etapy. W pierwszym poprzez estymację liniową wyznaczane są współczynniki przekształcenia perspektywicznego (liniowa transformata). Drugim etapem jest estymacja nieliniowa, w wyniku której w sposób iteracyjny wyznaczane są współczynniki nieliniowej części modelu zniekształceń.

Przyjęto kryterium optymalizacji opisane równaniem (3.38), które uwzględnia współrzędne środków markerów na obrazie i modelu:

$$J = \sum_{i=1}^n (U_i - u_i)^2 + \sum_{i=1}^n (V_i - v_i)^2, \quad (3.38)$$

gdzie  $(U, V)$  są współrzędnymi punktów (markerów) na obrazie źródłowym, a  $(u, v)$  to współrzędne wyliczone z modelu.

W wyniku przeprowadzonych optymalizacji na obrazach pochodzących z endoskopu *Olympus BF 160* wyznaczono następujące współczynniki wielomianu korekcyjnego:

$$r_k(r) = 4.2009 \cdot 10^{-8} r^4 + 1.5991 \cdot 10^{-10} r^3 + 3.7892 \cdot 10^{-13} r^2 + r. \quad (3.39)$$



Wyznaczony środek zniekształceń miał współrzędne pikselowe  $(x_0, y_0) = (188, 191)$ , zaś geometryczny środek obrazu znajdował się w punkcie o współrzędnych  $(191, 182)$ . Współrzędne środka zniekształceń wyrażone we współrzędnych unormowanych do rozmiarów obrazu mają wartość  $(0,4921, 0,5247)$ . Analizując otrzymane wyniki można zauważyć, że środek zniekształceń nie pokrywa się ze środkiem obrazu, co może sugerować, że sensor obrazu nie został umieszczony precyzyjnie w osi układu optycznego.

**Metoda uniwersalna** W celu weryfikacji wyników opracowanej metody dedykowanej użyto metody referencyjnej w postaci uniwersalnej biblioteki *Camera Calibration Toolbox for Matlab*, której autorem jest Jean-Yves Bouguet<sup>†</sup>. Biblioteka ta umożliwia wyznaczenie współczynników zarówno wewnętrznych, jak również zewnętrznych rozszerzonego modelu DLT (ang. *Direct Linear Transform*) kalibrowanej kamery. Model ten obejmuje następujące parametry kamery:

- $f_c$  – długość ogniskowej poziomej i pionowej wyrażona w pikselach zapisana w dwuelementowym wektorze,
- $cc$  – położenie punktu centralnego (ang. *principal point*), wyznaczającego środek układu współrzędnych obrazu i jednocześnie środek zniekształceń,
- $\alpha_c$  – kąt pomiędzy osiami układu współrzędnych oraz
- $kc$  – wektor pięciu współczynników zniekształceń radialnych.

Równanie (3.40) opisuje przyjęty model nieliniowych zniekształceń, który został zastosowany w bibliotece. Na współrzędne przesuniętego piksela obrazu mają wpływ dwa rodzaje zniekształceń: radialne (pierwsza część równania), zależne jedynie od promienia oraz tangencjalne (druga część równania):

$$\begin{bmatrix} x_{dist} \\ y_{dist} \end{bmatrix} = (1 + ar^2 + br^4 + cr^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2dxy & + e(r^2 + 2x^2) \\ d(r^2 + 2x^2) & + 2exy \end{bmatrix}, \quad (3.40)$$

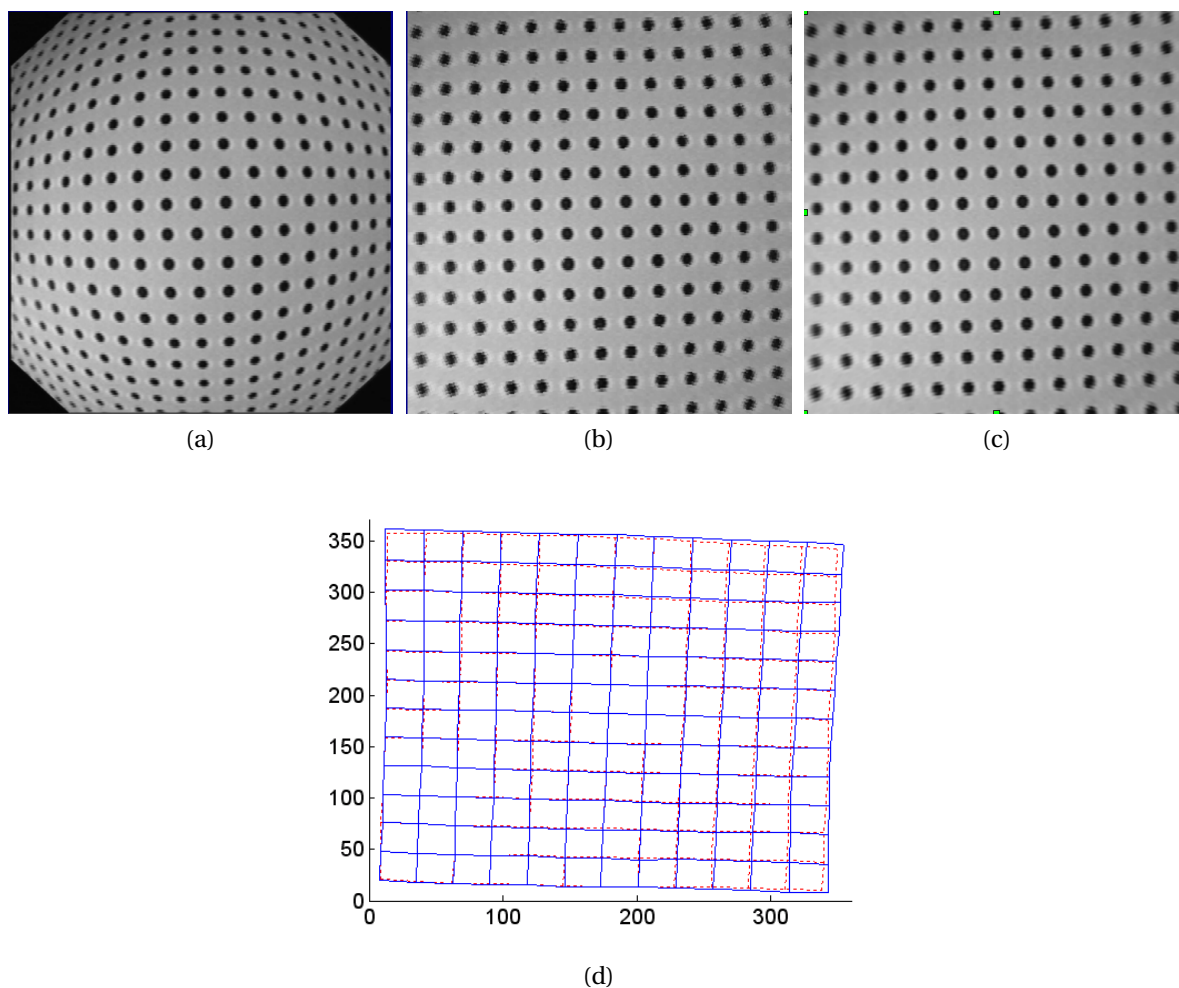
gdzie: parametry  $a, b, c$  odpowiadają za modelowanie zniekształceń radialnych, zaś  $d, e$  za współczynniki zniekształceń tangencjalnych, natomiast odległość  $r$  liczona jest od piksela  $(x, y)$  do środka zniekształceń  $(x_0, y_0)$ .

Identyfikacja parametrów wewnętrznych kamery polegała na wskazaniu na serii obrazów testowych narożników prostokątnego obszaru, na który składała się konkretna liczba „kratek”. Na tej podstawie algorytm generował punkty, które powinny znajdować się blisko narożników kratki na obrazie testowym, przy założonych wstępnych wartościach parametrów zniekształceń radialnych. Ostatnim krokiem, była analiza otoczenia punktów w poszukiwaniu dokładnej pozycji narożników. Na podstawie tak otrzymanych współrzędnych narożników, obliczane były ostateczne współczynniki modelu kamery.

**Porównanie wyników.** Na rysunku 3.18 przedstawiono porównanie wyników działania obydwu algorytmów w postaci obrazu zarejestrowanego przez endoskop (a), obrazów po skorygowanej dystorsji (b, c) oraz porównanie współrzędnych środków markerów (d).

Subiektywne porównanie wyników korekcji jedynie na podstawie obrazów jest trudne. Można jedynie zauważyć, że pole widzenia w metodzie dedykowanej, po wykonaniu korekcji, jest nieco większe. Do liczbowego ujęcia różnic pomiędzy metodami wyniki korekcji obydwu metod porównano w sensie średniokwadratowym, wykorzystując algorytm opracowany na potrzeby metody dedykowanej. Przez środki kół w skorygowanych obrazach testowych, poprowadzono linie poziome i pionowe, a następnie obliczono średniokwadratowy błąd dopasowania.

<sup>†</sup>Strona domowa: <http://www.vision.caltech.edu/bouguetj>



Rysunek 3.18: Wyniki korekcji zniekształceń: (a) obraz z endoskopu, (b) obraz skorygowany metodą dedykowaną i (c) uniwersalną oraz (d) porównanie pozycji środków kół (– metoda dedykowana, - - - metoda uniwersalna)

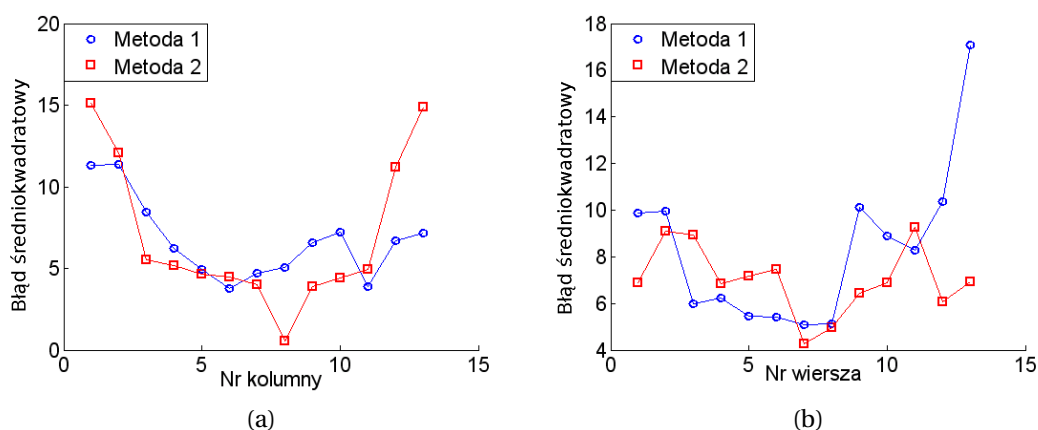
Tablica 3.1: Porównanie błędów średniokwadratowych dopasowania do prostych dla linii poziomych i pionowych na obrazach po korekcji (Socha i inni, 2005a)

Suma błędów średniokwadratowego	Metoda dedykowana	Metoda uniwersalna
kolumny	87.5	90.9
wiersze	107.8	91.2
kolumny i wiersze	195.3	182.1

Powstałe siatki przedstawiono na rysunku 3.18d. Wyniki liczbowe błędów dopasowania przedstawiono w tabeli 3.1. Dodatkowo, na rysunku 3.19 przedstawiono wartość błędów średniokwadratowych dopasowania dla obydwu metod w funkcji numeru kolumny (3.19a) oraz wiersza (3.19b). Można zauważyć, że błąd średniokwadratowy jest największy dla linii skrajnych, czyli najbardziej zniekształconych na obrazie oryginalnym.

Biorąc pod uwagę wyniki korekcji przedstawione na rysunkach 3.18 oraz uwzględniając wyniki liczbowe z tabeli 3.1 można stwierdzić, że obie metody oferują bardzo zbliżoną popraw-





Rysunek 3.19: Porównanie wartości błędu średniokwadratowego dopasowania linii do prostych: (a) dla wierszy oraz (b) kolumn

ność korekcji nieliniowych zniekształceń obrazu. Otrzymane wyniki korekcji dla obydwu metod są porównywalne. Żadna z metod nie przeprowadza korekcji znacząco lepiej zarówno w sensie średniokwadratowym, jak i na podstawie subiektywnej oceny wizualnej.

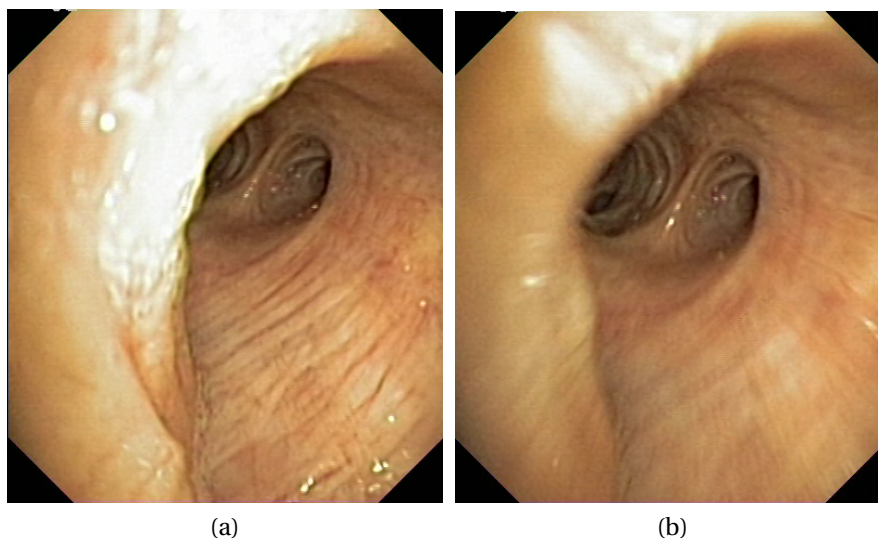
Pod względem funkcjonalności prostsza w aplikacji jest metoda dedykowana, dla której wystarczy zastosowanie jednego obrazu testowego, podczas gdy w metodzie drugiej konieczne jest zarejestrowanie kilku zdjęć testowych pod różnymi kątami. Metoda dedykowana jest w pełni automatyczna, natomiast metoda uniwersalna wymaga interakcji użytkownika w procesie estymacji zniekształceń. Z kolei metoda uniwersalna umożliwia nie tylko estymację i korekcję zniekształceń geometrycznych kamery, ale również identyfikację jej innych parametrów. Porównując wyniki obu metod można stwierdzić, że zaniedbanie zniekształceń tangencjonalnych w modelu metody dedykowanej jest uzasadnione.

### 3.3.3 Implementacja wirtualnego modelu kamery

Wirtualny model kamery endoskopu został zaimplementowany z wykorzystaniem biblioteki VTK i opiera się o model kamery otworkowej, która dokonuje przekształcenia perspektywicznego bez wprowadzania zniekształceń geometrycznych obrazu. Z tego względu, konieczne stało się uzupełnienie tego modelu o zjawiska charakterystyczne dla kamery endoskopowej (Socha i inni, 2009): zniekształcenia obrazu spowodowane szerokokątną optyką, rozmycie obrazu spowodowane skończoną głębią ostrości oraz rozmycie spowodowane gwałtownym ruchem. Przykłady tego typu zjawisk zaprezentowano na rysunku 3.20.

W przypadku endoskopów rozmyte są obiekty leżące najbliżej soczewki. Efekt rozmycia obrazu może być również wywołany przez gwałtowne ruchy końcówki endoskopu przy relatywnie długim czasie akwizycji obrazu. Wymienione zjawiska stanowią nieodzowny element rzeczywistych obrazów endoskopowych i powinny być uwzględnione w obrazach wirtualnych, generowanych na potrzeby szkolenia.

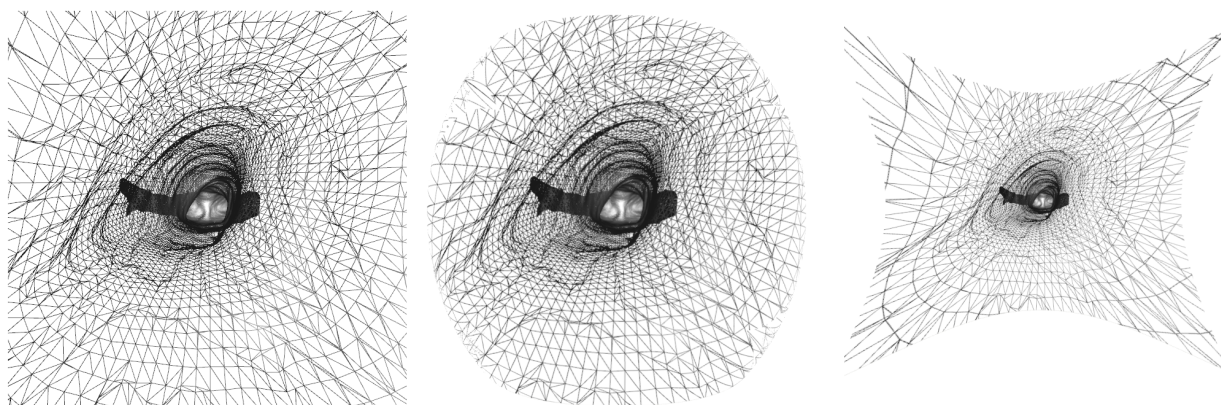
Za kontrolę parametrów wirtualnej kamery odpowiada klasa *vtkCamera*. Parametry zewnętrzne modelu kamery (wektor translacji oraz macierz rotacji) mogą być zadawane zarówno bezpośrednio, jak i pośrednio przez wskazanie punktu skupienia (ang. *focal point*). Ogniskowa sterowana jest w sposób pośredni, poprzez podanie kąta widzenia wirtualnej kamery. Dodatkowo, wirtualna kamera ma możliwość określenia przedniej i tylnej płaszczyzny odcięcia



Rysunek 3.20: Przykłady rozmycia rzeczywistego obrazu z endoskopu spowodowane: (a) skończoną głębią ostrości oraz (b) ruchem końcówki

(ang. *clipping plane*). Rozdzielczość wirtualnej kamery zależy od nastaw klasy *vtkRenderWindow*. Dystorsje obrazu oraz rozmycia zostały osiągnięte z wykorzystaniem dodatkowych filtrów umieszczonych w strumieniu wizualizacji.

**Dystorsja obrazu.** Obraz z endoskopu charakteryzuje się zniekształceniami beczkowymi, które można modelować wielomianem. Implementacja zniekształceń geometrii obrazów wirtualnych wymaga wykonania kosztownych obliczeniowo operacji interpolacji. Z tego względu zdecydowano się wykorzystać moc obliczeniową karty graficznej. Wygenerowany obraz wirtualnej endoskopii (bez zniekształceń) traktowany jest jako tekstura, która następnie jest nakładana na płaszczyznę rozpiętą na regularnej siatce punktów. Położenie punktów opisuje zniekształcenie wprowadzane przez optykę endoskopu. Przykłady wirtualnych obrazów przed i po uwzględnieniu zniekształceń przedstawiono na rysunku 3.21. Ze względu na lepszą czytelność, wszystkie wizualizacje wirtualnej bronchoskopii przedstawiają jedynie krawędzie łączące poszczególne wierzchołki zrekonstruowanej powierzchni tchawicy.



Rysunek 3.21: Symulowane zniekształcenia obrazu: (a) obraz oryginalny, (b) zniekształcenia beczkowe oraz (c) poduszkowe

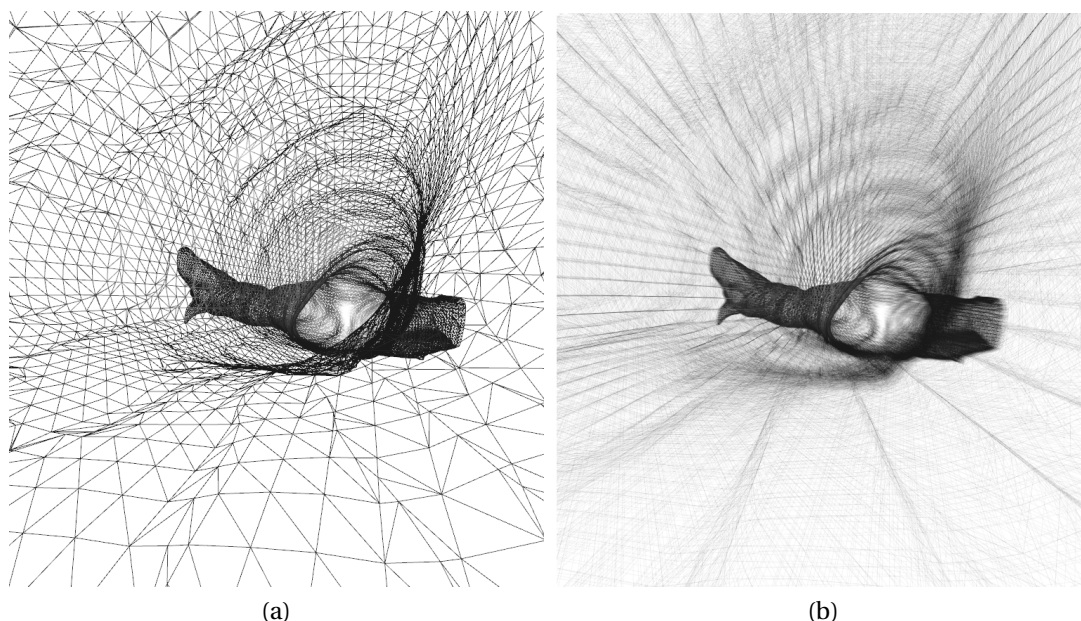
Możliwa jest również inna metoda wprowadzania zniekształceń, polegająca na bezpośredniej modyfikacji przestrzennej siatki trójkątów, tworzących powierzchnię przed jej narysowaniem. Zamiast przesuwac piksele na wygenerowanym obrazie deformowana jest cała bryła. Współrzędne wierzchołków przesuwane są prostopadłe do osi kamery i uwzględniają odległość od kamery oraz odległość wierzchołka od prostej wyznaczonej przez współrzędne wirtualnej kamery i środek zniekształceń. Konieczne jest przeliczenie nowych przesunięć wszystkich wierzchołków przy każdej zmianie pozycji kamery. Ze względu na bardzo dużą liczbę wierzchołków występujących w rekonstruowanych modelach drzewa oskrzelowego, metoda ta zaimplementowana z wykorzystaniem jedynie biblioteki VTK okazała się zbyt wolna, by mogła być zastosowana w systemach działających interaktywnie.

**Rozmycie obrazu.** Ograniczona ilość światła oświetlającego przestrzeń przed endoskopem powoduje konieczność stosowania długich czasów akwizycji obrazu. Długie naświetlanie sensora ma również pozytywny wpływ na jakość obrazu – powoduje ograniczenie szumów w obrazie. Z drugiej strony, podczas ruchu końcówki endoskopu obraz zostaje rozmyty (rysunek 3.20). Rozmycie spowodowane ruchem może być dobrze symulowane poprzez akumulację wielu obrazów. Dla każdego obrazu obliczana jest nowa pozycja przestrzenna kamery, uwzględniająca przesunięcie oraz obrót i wynikająca z ruchu między dwoma pozycjami w przestrzeni. Pozycje te wyznaczają początek oraz koniec symulowanej akwizycji obrazu. Takie podejście jest zgodne z rzeczywistym źródłem powstawania zniekształceń – w skończonym czasie, koniecznym do zarejestrowania obrazu, następuje przesunięcie kamery. Jeżeli przesunięcie jest niewielkie, efekt rozmycia jest niewidoczny. Na jakość modelowanego rozmycia ma znaczący wpływ liczba akumulowanych obrazów. Liczba ta może być dobierana w sposób adaptacyjny, np. na podstawie odległości jaką ma pokonać kamera. Na rysunku 3.22 przedstawiono wyniki działania algorytmu akumulacji obrazów, który umożliwia symulację rozmycia obrazu spowodowanego ruchem końcówki endoskopu oraz długiego czasu akwizycji obrazu (ang. *motion blur*).

Drugim powodem występowania rozmycia na obrazach endoskopowych jest skończona głębia ostrości. Głębnię ostrości definiuje się jako zakres odległości, w którym postrzegany obraz nie jest rozmyty. Z fizycznego punktu widzenia w torze optycznym istnieje tylko jeden punkt, w którym obraz jest ostry (skupiany). Obraz poza ogniskiem nie jest punktem, a jego kształt definiowany jest przez tzw. krążek rozmycia. Ze względu na skończoną przestrzenną rozdzielczość sensorów optycznych, istnieje pewien obszar wokół punktu skupienia również postrzegany jako ostry. W tym obszarze krążek rozmycia jest mniejszy niż wymiar detektora i dlatego postrzegany jest jako punkt.

Na zakres głębi ostrości mają wpływ takie parametry jak: długość ogniskowej, wielkość przesłony czy odległość, na którą ustalona jest ostrość. W endoskopach można spodziewać się układów optycznych o krótkiej ogniskowej (soczewka szerokokątna). Przesłona nie jest w nich regulowana i ustalona jest na jedną wartość, która wynika zazwyczaj z konstrukcji. Stosowany światłowód optyczny ma skończoną średnicę i może zostać potraktowany jako przesłona. Również ostrość ustalona jest na jedną odległość, zazwyczaj dobraną do konkretnych zastosowań. Z przeprowadzonych obserwacji rzeczywistych obrazów pochodzących z wideobronchofibroskopu można wnioskować, że najbardziej prawdopodobnym zakresem ostrości jest wartość od kilku milimetrów do kilkunastu centymetrów. Taki przedział dobrany jest do wymiarów obserwowanych struktur drzewa oskrzelowego.

Implementacja głębi ostrości w obrazach wirtualnych może być wykonana na dwa sposoby:



Rysunek 3.22: Wyniki symulacji rozmycia obrazu spowodowanego szybkim ruchem (ang. *motion blur*) końcówki endoskopu: (a) obraz oryginalny oraz (b) obraz z dodanym efektem rozmycia przy ruchu „na wprost”

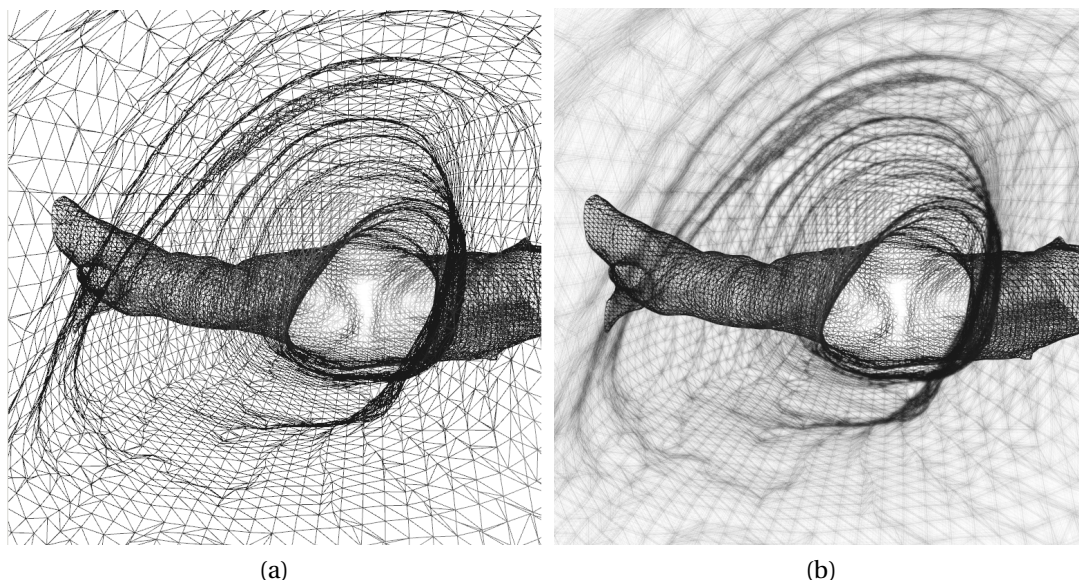
- poprzez akumulację wielokrotnie generowanych obrazów, dla których kamera jest przemieszczana o niewielką odległość wokół zadanej pozycji, co przy dużej liczbie sumowanych obrazów symuluje rzeczywisty krążek rozmycia,
- na drodze modyfikacji pikseli jednokrotnie wygenerowanego obrazu z uwzględnieniem ich odległości od płaszczyzny obrazu.

Przykład obrazu wirtualnego z zasymulowanym efektem skończonej głębi ostrości toru optycznego przedstawiono na rysunku 3.23. Dla każdego użytego obrazu, kamera jest obracana wokół punktu skupienia promieni świetlnych tzw. *focal point*, umieszczonego przed kamerą. Punkt ten wyznacza kierunek „patrzenia” wirtualnej kamery i może być wykorzystany do określania punktu ostrości. W przypadku akumulowania obrazów niezbędne jest również określenie liczby obrazów oraz maksymalnej odległości, o jaką będzie przesuwana kamera. Przy niewielkim rozmyciu obrazu wystarczą kilkanaście obrazów. Do uzyskania naturalnego rozmycia dla bardzo wąskiej głębi ostrości, niezbędna jest akumulacja kilkudziesięciu obrazów, co nie pozostaje bez wpływu na wydajność generowania kompletnego obrazu.

### 3.4 Model oświetlenia endoskopu

Ostatnim elementem opracowanego modelu zabiegu wirtualnej bronchoskopii jest oświetlenie. W rzeczywistym endoskopie stosuje się jedno lub dwa źródła światła w postaci światłowodów podłączonych do modułu kontroli oświetlenia endoskopu. Zadaniem tego modułu jest dostarczanie odpowiedniej ilości światła, którego moc dobierana jest w sposób adaptacyjny, w zależności od aktualnej sytuacji w polu zabiegowym endoskopu. Wiązka światła uformowana jest w taki sposób, by nie powodowała prześwieclenia najbliższych ścianek, a jednocześnie, by dobrze oświetlała oddalone fragmenty drzewa oskrzelowego, znajdujące się na wprost





Rysunek 3.23: Wyniki symulacji skończonej głębi ostrości: obraz oryginalny (a) oraz dodana głębia ostrości (b)

końcówki. Autor zrezygnował z modelowania dwóch źródeł światła, występujących w rzeczywistym endoskopie z dwóch powodów. Po pierwsze, w obrazach rzeczywistych światła są nierozróżnialne. W obrazach nie występują np. podwójne rozbłyski światła, które mogłyby świadczyć o występowaniu dwóch źródeł światła. Dodatkowo, uwzględnienie dwupunktowego światła zmniejszyłoby wydajność generowania obrazów oraz wprowadziło artefakty.

W opracowanym modelu oświetlenie endoskopu zaimplementowane zostało przy użyciu pojedynczego źródła światła kierunkowego, którego pozycja pokrywała się z pozycją kamery. Dodatkowo, światło było obracane wraz z kamerą, tak by zawsze oświetlało przestrzeń przed kamerą. W zastosowanym modelu oświetlenia uwzględniono zanik światła wraz z odległością, który został obliczony według następującej zależności:

$$k_{attenuation} = \frac{1}{k_c + k_l d + k_q d^2}, \quad (3.41)$$

gdzie  $d$  jest odległością od źródła światła, zaś współczynniki  $k_c$ ,  $k_l$  i  $k_q$  odpowiadają za stałą, liniowy oraz kwadratowy współczynnik zaniku światła. Dodatkowe współczynniki są niezbędne, by uwzględnić ograniczoną przestrzeń w tchawicy, która powoduje wielokrotne odbicia światła i doświetlanie odległych fragmentów drzewa oskrzelowego.

Drugim ważnym zaimplementowanym efektem jest zanik stożka światła. Dzięki temu obiekty znajdujące się na wprost kamery są oświetlone najbardziej, zaś te, które znajdują się na skraju obrazu oświetlone są słabiej. Parametrami tego efektu są dwa kąty, które definiują wewnętrzny stożek światła o stałym natężeniu światła oraz zewnętrzny stożek światła, w którym natężenie oświetlenia maleje od wartości maksymalnej do zera. Przejście od wartości maksymalnej do minimalnej zaimplementowano z wykorzystaniem gładkiej interpolacji Hermite'a:

$$k_{smooth}(x_{min}, x_{max}, x) = -2 \cdot \left( \frac{x - x_{min}}{x_{max} - x_{min}} \right)^3 + 3 \cdot \left( \frac{x - x_{min}}{x_{max} - x_{min}} \right)^2. \quad (3.42)$$

Zastosowany model oświetlenia nierozzerwalnie łączy się z opracowanym modelem oświetlenia, w którym to dla każdego piksela obrazu wyliczane są wszystkie współczynniki oświetlenia oraz uwzględniane parametry materiału (odbicia i rozproszenia).

### 3.5 Podsumowanie

Autor rozprawy opracował metodologię budowy modelu zabiegu bronchoskopowego, który obejmuje: proceduralne tworzenie kształtu i wyglądu drzewa oskrzelowego, uwzględnienie nieliniowych zjawisk zniekształcających obraz, a występujących w rzeczywistym instrumencie medycznym oraz uwzględnienie występującego w rzeczywistości oświetlenia. Opracowane i przetestowane w praktyce metody umożliwiają generowanie obrazów wirtualnej bronchoskopii o różnych właściwościach. Możliwe jest zarówno tworzenie bardzo prostych (tubowych) struktur anatomicznych, jak również wersji skomplikowanych, uwzględniających potrzebne szczegóły morfologiczne. Opracowany model drzewa oskrzelowego nadaje się więc zarówno do testowania, jak i weryfikacji algorytmów estymacji ruchu endoskopu. Dzięki zaawansowanym możliwościom sterowania wyglądem powierzchni generowanego drzewa oskrzelowego, możliwe jest wykorzystanie go również w charakterze fantomu szkoleniowego.

Zastosowany przez autora rozprawy model oświetlenia jest w pełni parametryzowalny. W najbardziej zaawansowanej i rozbudowanej wersji jest zmodyfikowanym pod kątem wydajności oraz uwzględniającym tekstuowanie modelem odbicia rozproszonego Oren-Nayar'a. Dodano do niego również odbicie zwierciadlane obliczane metodą Blinna-Phonga oraz efekt Fresnela aproksymowany funkcją Schlicka. Wadą tak rozbudowanego modelu oświetlenia jest znaczne zapotrzebowanie na moc obliczeniową karty graficznej, powodujące zauważalny spadek wydajności generowanego obrazu.

## METODA ŚLEDZENIA KOŃCÓWKI BRONCHOFIBEROSKOPU

Opracowana metoda wspomagania zabiegów bronchofiberoskopii opiera się na danych z tomografii komputerowej oraz obrazie wideo z bronchoskopu. Dane z tomografii komputerowej służą do rekonstrukcji trójwymiarowej powierzchni drzewa oskrzelowego pacjenta. Na jej podstawie generowane są obrazy wirtualnej bronchoskopii, które są następnie porównywane z obrazem rzeczywistym w celu określenia pozycji rzeczywistego endoskopu. Jeżeli obraz wirtualny jest „wystarczająco podobny” do rzeczywistego obrazu z bronchoskopu, uznaje się, że pozycja endoskopu rzeczywistego jest taka sama jak endoskopu wirtualnego.

### 4.1 Przegląd metod

Sekwencje dwuwymiarowych obrazów (rejestrowanych przez kamerę bądź sztucznie generowanych) mogą być analizowane w celu oceny ruchu występującego na tych obrazach. Analiza ta wykonywana jest w różnym celu i może być wykonywana różnymi metodami. Do podstawowych zadań analizy ruchu można zaliczyć:

- wyznaczenie *ruchu własnego kamery* (ang. *egomotion*) — analiza ma na celu określenie *trójwymiarowej sztywnej transformacji*, która opisuje ruch kamery względem statycznej sceny i obejmuje przesunięcie oraz obrót kamery względem otoczenia,
- *śledzenie ruchu obiektów* (ang. *tracking*) — w wyniku analizy otrzymywane są wektory ruchu fragmentów obrazu, którymi mogą być obiekty (np. ludzie, samochody), poruszające się na tle ruchomej lub nieruchomej sceny,
- wyznaczanie *przepływu optycznego* (ang. *optical flow*) — dla każdego punktu obrazu wyznaczany jest jego wektor ruchu w płaszczyźnie obrazu, który obejmuje zarówno ruch obserwowanej sceny, jak i ruch kamery; informacja taka wykorzystywana jest między innymi do kompresji obrazów.

Problem śledzenia końcówki endoskopu w drzewie oskrzelowym, jedynie na podstawie obrazu rejestrowanego przez endoskop, jest klasycznym problemem rozpoznawania *ruchu własnego* (ang. *egomotion*) i stanowi jedno z fundamentalnych zagadnień „widzenia maszynowego” (ang. *machine vision*) (Tian i inni, 1996; Hartley i Zisserman, 2004). W odróżnieniu od da-



nych z endoskopu, w systemach nawigacji używanych w robotyce stosuje się zazwyczaj układy wielokamerowe (Cyganek, 2002): stereowizyjne zapewniające dodatkową informację o „głębi” obrazu (np. odległości do przeszkody lub ściany) lub dookolne (ang. *omnidirectional camera*), które rejestrują szerokokątny panoramiczny obraz wokół robota, będący zazwyczaj wycinkiem sfery.

Metody estymacji ruchu własnego można podzielić na dwie główne grupy: metody pośrednie oraz bezpośrednie. W metodach pośrednich do wyznaczenia ruchu kamery wykorzystywane są elementy charakterystyczne (ang. *feature-based method*), takie jak linie lub punkty (tzw. znaczniki, np. narożniki, przecięcia linii itp.). W metodach tych następuje znaczna redukcja analizowanych danych — z całego zbioru pikseli wybierany jest jedynie niewielki podzbiór, który zapewnia uzyskanie największej ilości informacji. Następnie, na podstawie analizy ruchu niewielkiego zbioru znaczników wyznaczany jest ruch kamery. Metody te, pomimo znacznej redukcji danych, są nadal kosztowne obliczeniowo, ponieważ wymagają dodatkowych obliczeń potrzebnych na znalezienie znaczników. Z kolei metody bezpośrednie bazują bezpośrednio na analizie przepływu optycznego obrazu (Daniilidis, 1997).

Przepływ optyczny definiowany może być jako relacja pola ruchu, rozumianego jako dwuwymiarowa projekcja obserwowanych przez kamerę obiektów, do występujących na obrazie przesunięć grup pikseli (Bradski i Davis, 2002). Zakładając, że jasność obrazu jest stała, przepływ optyczny można zapisać przy pomocy następującej zależności:

$$I(x, y, t) = I(x + u, y + v, t + 1), \quad (4.1)$$

gdzie:  $I = I(x, y, t)$  jest jasnością obrazu, będącą funkcją współrzędnych obrazu  $(x, y)$  i czasu  $t$ , zaś  $(u, v)$  są przesunięciami pikseli.

Inną definicję przepływu optycznego można znaleźć w pracy (Trucco i Verri, 1998). Autorzy określają w niej przepływ optyczny jako dwuwymiarowe pole wektorowe, które opisuje pozorny ruch fragmentów obrazu objawiający się zmianami jasności grup pikseli. Obie definicje są podobne i zakładają, że jasność obrazu jest stała. Zakładając dodatkowo różniczkowalność obrazu (czasową i przestrzenną) warunek ten można zapisać jako:

$$\frac{dI}{dt} = 0. \quad (4.2)$$

Przy takich założeniach, współrzędne obrazu  $x$  i  $y$  stają się funkcjami czasu, co pozwala zapisać równanie (4.2) w następującej postaci:

$$\frac{dI(x(t), y(t), t)}{dt} = \frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0. \quad (4.3)$$

Podstawiając za  $\frac{\partial x}{\partial t} = u$  oraz  $\frac{\partial y}{\partial t} = v$ , otrzymujemy dwuwymiarowe pole przepływu zdefiniowane wektorem postaci  $\mathbf{v} = (u, v) = \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}\right)$ , co pozwala uprościć zapis równania przepływu:

$$\frac{dI(x(t), y(t), t)}{dt} = \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} = 0. \quad (4.4)$$

Pochodną jasności obrazu o zdefiniowanych osiach  $x$  i  $y$  jest gradient obrazu  $\nabla I$ . Przyjmując dodatkowo, że  $u$  i  $v$  są składowymi pola przepływu, równanie (4.4) można zapisać jako:

$$(\nabla I)^T \mathbf{v} + I_t = 0, \quad (4.5)$$

gdzie:  $I = I(x, y, t)$  jest jasnością obrazu,  $\mathbf{v}$  jest dwuwymiarowym polem przepływu, zaś  $t$  oznacza różniczkę cząstkową w danej chwili czasowej. Zmieniając sposób oznaczania czasu, równanie (4.5) można zapisać w postaci:

$$I_t = (\nabla I_{t+1})^T \mathbf{v}, \quad (4.6)$$

którą można porównać z równaniem (4.1). Można zauważyć, że obydwa równania (4.1) i (4.6) odpowiadają sobie, co dowodzi, że przytoczone definicje przepływu optycznego są równorzędne.

Do najpopularniejszych metod obliczania przepływu optycznego można zaliczyć następujące metody:

- blokowe, bazujące głównie na minimalizacji sum kwadratów różnic obrazu oraz korelacji fragmentów obrazu,
- szacowania przepływu optycznego, bazujące na pochodnych cząstkowych, np. metody Lucas-Kanade (Lucas i Kanade, 1981), Horn-Schuncka (Horn i Schunck, 1981), Buxton (Murray i Buxton, 1987), Black-Jepson (Beauchemin i Barron, 1995) lub ogólne metody wariacyjne.

Metody stosowane w robotyce, zwłaszcza opierające się na punktach charakterystycznych, mają ograniczone zastosowanie w bronchoskopii. Głównie ze względu na charakter obrazów medycznych – nie występują na nich punkty charakterystyczne, takie jak krawędzie spotykane w obrazach „niemedycznych“.

W celu weryfikacji tej tezy, przetestowano jedną z „klasycznych” metod szacowania przepływu optycznego, jaką jest algorytm Lucas-Kanade (Lucas i Kanade, 1981). Algorytm ten sprowadza różniczkowy opis przepływu optycznego do równań liniowych. Zakłada się, że przepływ optyczny  $\mathbf{v}$  jest stały w niewielkim otoczeniu punktu  $P(x, y)$ . Otoczenie to ma postać okna o wymiarach  $m$  na  $m$  pikseli ( $m > 1$ ), zaś punkt  $P$  znajduje się w centrum tego okna. Dla tak przyjętego otoczenia należy wyznaczyć  $n = m^2$  równań przepływu:

$$\begin{aligned} I_{x_1} u + I_{y_1} v &= -I_{t_1} \\ I_{x_2} u + I_{y_1} v &= -I_{t_1} \\ &\vdots \\ I_{x_n} u + I_{y_n} v &= -I_{t_n}, \end{aligned} \quad (4.7)$$

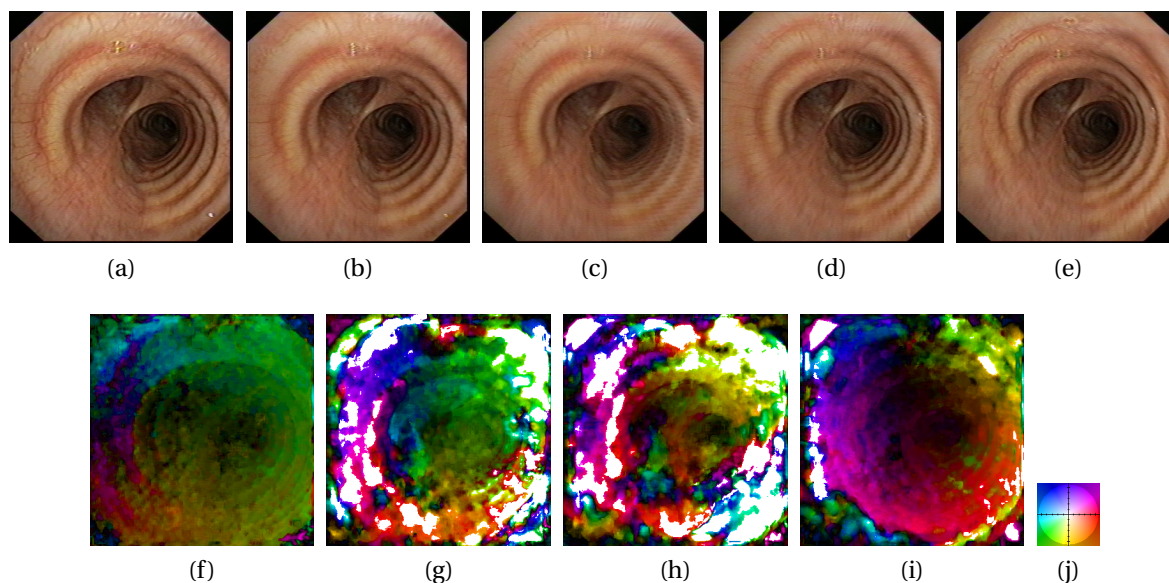
gdzie:  $u$  i  $v$  są składowymi prędkościami ruchu  $\mathbf{v} = (u, v)$ , zaś  $I_x = \frac{\partial I}{\partial x}$ ,  $I_y = \frac{\partial I}{\partial y}$  oraz  $I_t = \frac{\partial I}{\partial t}$  są pochodnymi obrazu w konkretnych kierunkach. W zapisie macierzowym równanie (4.7) przyjmuje następującą postać:

$$\begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_{t_1} \\ -I_{t_2} \\ \vdots \\ -I_{t_n} \end{bmatrix}. \quad (4.8)$$

Do wyznaczenia składowych wektora ruchu używana jest metoda najmniejszych kwadratów:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_{x_i}^2 & \sum I_{x_i} I_{y_i} \\ \sum I_{x_i} I_{y_i} & \sum I_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_{x_i} I_{t_i} \\ -\sum I_{y_i} I_{t_i} \end{bmatrix}. \quad (4.9)$$

Rysunek 4.1 przedstawia przykład zastosowania algorytmu Lucas-Kanade w wersji wielorozdzielczej do wyznaczenia przepływu optycznego na obrazie z wideoendoskopu (Marzat,



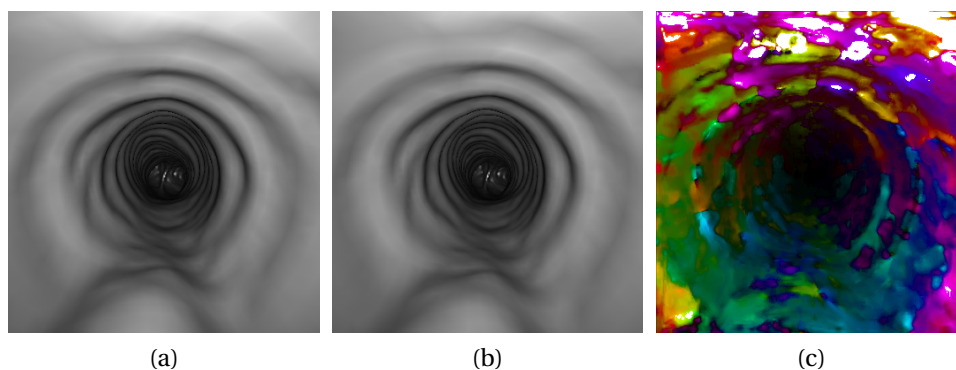
Rysunek 4.1: Przepływ optyczny (f-i) wyznaczony metodą Lucas-Kanade dla kolejnych klatek filmu z wideoendoskopu (a-e), kierunek przepływu został zakodowany kolorem zgodnie z diagramem (j)

2009). Dla pięciu kolejnych obrazów (rysunek 4.1a-e) obliczony przepływ zakodowano w postaci kolorów (rysunek 4.1j). Jasność koloru reprezentuje moduł wektora przepływu (kolor ciemny to mały przepływ, kolory intensywne to duże przesunięcia).

Można zauważyć, że przepływ wyznaczony dla obrazów endoskopowych zawiera wiele obszarów ścian drzewa oskrzelowego (położonych blisko siebie) o bardzo zróżnicowanych kierunkach ruchu. Przykładowo rysunek 4.1f przedstawia niewielki ruch między klatkami 4.1a i 4.1b, na którym widoczny jest „szum” zarówno kierunku, jak i wielkości przesunięcia. Gdy wielkość przesunięcia się zwiększa (ramki 4.1b-d) wyniki obliczonego przepływu optycznego są bardzo różnorodne, co świadczy jednocześnie o dużej czułości algorytmu przy jednoczesnym generowaniu informacji trudnej do dalszej, szybkiej analizy. Algorytm Lucas-Kanede przetestowano również na obrazach wirtualnych. Wynik przedstawiono na rysunku 4.2. Przepływ optyczny wyznaczony dla obrazów wirtualnych pozbawionych tekstury jest jeszcze gorszy do dalszej analizy – jest bardzo zaszumiony i posiada bardzo liczne artefakty, które dyskwalifikują tę metodę do szacowania ruchu własnego endoskopu.

Czas obliczania przepływu dla dwóch obrazów o rozdzielczości 360 na 360 pikseli przekraczał 10 sekund (środowisko Matlab, procesor o częstotliwości 2,4GHz), co uniemożliwia zastosowanie metody do obliczeń w czasie rzeczywistym. Istnieją wprawdzie szybsze implementacje tego algorytmu (również w CUDA), jednak ze względu na zaszumienie wyników, są one nadal mało użyteczne. Przepływ optyczny wyznaczony „klasyczną”, dla automatyki i robotyki metodą Lucas-Kanede, zdaniem autora rozprawy, nie jest użyteczny do szybkiego estymowania położenia końcówki endoskopu.

W literaturze można znaleźć przykłady użycia śledzenia punktów charakterystycznych na dwóch obrazach za pomocą geometrii epipolarnej (Mori i inni, 2002) oraz z wykorzystaniem przepływu optycznego (ang. *optical-flow*) (Helferty i Higgins, 2002). Badania w tym kierunku opisano w publikacji (Zieliński i inni, 2006). Algorytmy zaprojektowane na potrzeby robotyki zwykle zakładają „sztywność” analizowanej sceny (ang. *rigid environment*). W przypadku



Rysunek 4.2: Przepływy optyczne (c) wyznaczone metodą Lucas-Kanade dla obrazów wirtualnej bronchoskopii (a-b), przy ruchu kamery „na wprost”

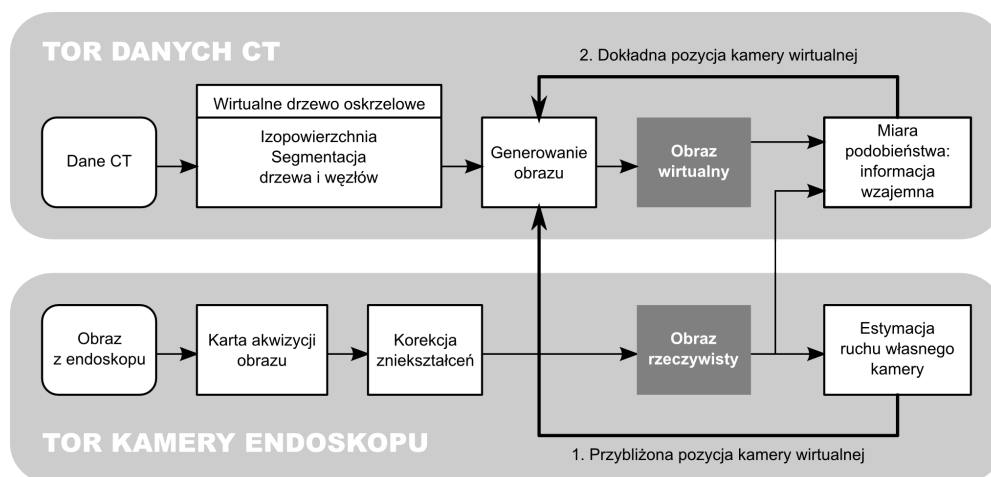
drzewa oskrzelowego nie można mówić o spełnieniu tego warunku, ponieważ w obrazie występują liczne ruchy niezwiązane z przemieszczaniem się endoskopu, np. spowodowane oddychaniem, odkształcaniem wydzielin czy biciem serca. Dodatkowym czynnikiem utrudniającym analizę obrazów jest adaptacyjne oświetlenie o zmiennym natężeniu oraz często występujące zanieczyszczenia na przedniej soczewce. Algorytmy tego typu są bardzo podatne na zakłócenia i generują zaszumione wektory ruchu.

Do śledzenia pozycji końcówki endoskopu mogą być też użyte metody oparte na rekonstrukcji trójwymiarowej powierzchni drzewa oskrzelowego na podstawie oświetlenia (ang. *shape-from-shading*) (Bricault i inni, 1998) lub poprzez triangulację punktów charakterystycznych (Deligianni i inni, 2003). Uzyskane w ten sposób powierzchnie są następnie dopasowywane do danych uzyskanych z tomografii komputerowej. Wyniki prac nad tymi metodami zaprezentowano w publikacji (Bułat i inni, 2006).

## 4.2 Opracowana metoda śledzenia położenia końcówki bronchoskopu

Rysunek 4.3 przedstawia schemat blokowy opracowanego modelu komputerowego systemu wspomagania zabiegu biopsji aspiracyjnej cienkoigłowej (BAC), wykonywanej w trakcie bronchoskopii (Zieliński i inni, 2007; Bułat i inni, 2010). System ten składa się z dwóch torów przetwarzania danych. Pierwszy służy do przetwarzania danych z tomografii komputerowej, w celu szybkiego generowania obrazów wirtualnej bronchoskopii. Natomiast drugi tor służy do przetwarzania obrazu wideo pochodzącego z kamery endoskopowej.

Do poprawnej pracy systemu potrzebne są dane z wykonanej przed zabiegiem tomografii komputerowej. Na podstawie tych danych rekonstruowana jest powierzchnia drzewa oskrzelowego z wykorzystaniem opracowanego programu (Socha i inni, 2005b) oraz biblioteki VTK (Schroeder i inni, 2004). Powierzchnia drzewa oskrzelowego wyznaczana jest jako izopowierzchnia przy użyciu algorytmu *marching cube* (Lorensen i Cline, 1987). Następnie jest ona optymalizowana, decymowana z zachowaniem kształtu, a w końcowym etapie zapisywana do pliku w postaci siatki połączonych trójkątów. Z przeprowadzonych eksperymentów wynika, że izopowierzchnia o wartość  $-500$  HU odpowiada granicy między powietrzem a tkankami tchawicy. Taka wartość zapewnia rekonstrukcję powierzchni drzewa oskrzelowego bez „dziur”,



Rysunek 4.3: Schemat blokowy opracowanego modelu systemu nawigacji, wspomagającego pozycjonowanie końcówki bronchofiberoskopu (Zieliński i inni, 2007)

które występują przy wyższej wartości HU, oraz z poprawnie odtworzonymi szczegółami anatomicznej budowy tchawicy.

Opracowane oprogramowanie umożliwia przeglądanie danych radiologicznych w postaci prezentacji dwuwymiarowych oraz trójwymiarowych. Dzięki temu, lekarz pulmonolog może w prosty i intuicyjny sposób zapoznać się z budową anatomiczną konkretnego pacjenta oraz zaplanować zabieg. Oprogramowanie dysponuje narzędziami do przeglądania danych oraz do pomiarów gęstości radiologicznej tkanek. Funkcje te umożliwiają odszukanie powiększonych węzłów chłonnych, znajdujących się w bezpośredniej okolicy drzewa oskrzelowego. Trójwymiarowa rekonstrukcja tchawicy oraz węzłów chłonnych ułatwia zaplanowanie dogodnego miejsca do wykonania biopsji. Miejsce takie może być oznaczane markerem i w trakcie zabiegu łatwo zidentyfikowane.

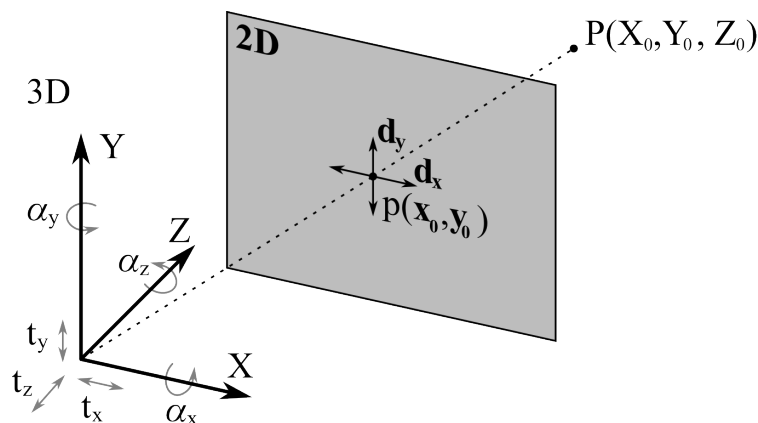
Podczas zabiegu bronchoskopii lekarz pulmonolog wprowadza endoskop z kamerą do drzewa oskrzelowego pacjenta. Przemieszcza endoskop w taki sposób, by trafić igłą w węzeł chłonny i pobrać z niego tkanki do dalszej analizy. W pierwszej wersji oprogramowania wspomagającego (Socha i Duplaga, 2005) lekarz kierował się na marker porównując obraz wyświetlany na monitorze komputera z obrazem z endoskopu. Natomiast w opracowanej wersji systemu wspomagania zabiegu, obraz z endoskopu zostaje rejestrowany i korygowane są jego nieliniowe zniekształcenia występujące w szerokokątnym torze optycznym kamery (tzw. efekt *rybiego oka*, rozdział 3.3). Kolejnym krokiem jest dwuetapowa (zgrubna i dokładna) estymacja parametrów trójwymiarowego przesunięcia endoskopu względem tchawicy. Pierwsza „zgrubna” metoda służy do szybkiego, ale przybliżonego oszacowania ruchu endoskopu: przesunięcia w przód i w tył oraz obrotu wokół własnej osi. Metoda ta, oznaczona na rysunku 4.3 blokiem *Estymacja ruchu własnego kamery*, bazuje na różnicach jakie występują w obrazie kolejnych klatek rzeczywistego obrazu wideo pochodzącego z endoskopu. Dokładny opis metody znajduje się w rozdziale 4.3. W wyniku jej działania ustalana jest wstępna pozycja kamery wirtualnej. Zadaniem drugiej „dokładnej” metody jest uwzględnienie pozostałych parametrów ruchu kamery endoskopu. Metoda ta bazuje na analizie informacji wzajemnej obrazów wirtualnego i rzeczywistego. Obraz wirtualny generowany jest wielokrotnie, przy zmienianych parametrach wirtualnej kamery w poszukiwaniu lepszego dopasowania obrazu rzeczywistego i wirtualnego (maksymalizacja informacji wzajemnej między nimi).

W celu dalszego przyśpieszenia procesu śledzenia ruchu końcówki endoskopu opracowano szybką i prostą metodę przybliżonej estymacji przesunięcia i obrotu kamery (Twardowski i inni, 2006). W metodzie tej zamodelowano otoczenie końcówki endoskopu w postaci cylindra oraz uwzględniono fakt występowania rozbłysku na rozwidleniu głównym (karinie). Następnie dla sąsiednich klatek filmu „rozwinęto” cylinder poprzez dokonanie transformacji obrazu do układu biegunowego o środku w wyznaczonym rozbłysku i skorygowano perspektywę. W kolejnym kroku piksele obrazu korelowano: wzdłuż promienia, otrzymując przesunięcie kamery, oraz wzdłuż promienia, otrzymując kąt obrotu kamery. Przeprowadzone testy potwierdziły szybkość prezentowanej metody oraz zadowalającą poprawność estymowanych parametrów przesunięcia i obrotu.

Model całego systemu wspomagania zabiegu biopsji został najpierw zaimplementowany w języku Matlab. System ten testowano mając do dyspozycji zapis wideo zabiegu bronchoskopii oraz dane z tomografii komputerowej pacjenta. Pobierając kolejne klatki filmu dopasowywano obraz z kamery wirtualnej, której pozycja przestrajana była według algorytmu zaprezentowanego na rysunku 4.3. Z przeprowadzonych badań wynika, że możliwe jest śledzenie końcówki endoskopu, jednak ze względu na złożoność obliczeniową, algorytm ten nie był w stanie pracować w czasie rzeczywistym (Zieliński i inni, 2007).

#### 4.2.1 Analiza ruchu kamery

Rysunek 4.4 przedstawia wpływ ruchu kamery na rzut punktu  $P$  o współrzędnych  $(X_0, Y_0, Z_0)$ , znajdującego się w trójwymiarowej przestrzeni.



Rysunek 4.4: Ruch kamery w przestrzeni 3D (translacja  $t$  oraz rotacja  $\alpha$ ) powodują zmianę współrzędnych  $(d_x, d_y)$  rzutu punktu  $P(X_0, Y_0, Z_0)$  na dwuwymiarowej płaszczyźnie obrazu

Zmianę współrzędnych punktu  $P(X_0, Y_0, Z_0)$  opisuje następujące przekształcenie (Bułat i inni, 2006):

$$\mathbf{P}' = \mathbf{R}\mathbf{P} + \mathbf{T}, \quad (4.10)$$

gdzie  $R$  jest macierzą rotacji, zaś  $T$  jest macierzą translacji. Dla obrotów o niewielki kąt  $\alpha(\alpha_x, \alpha_y, \alpha_z)$ , przyjmując kartezjański układ współrzędnych, można przyjąć następujące założenia:

$$\begin{aligned} \sin(x) &\approx x, \\ \cos(x) &\approx 1. \end{aligned} \quad (4.11)$$

Uwzględniając powyższe uproszczenia, macierz rotacji ma postać:

$$\mathbf{R} = \begin{bmatrix} 1 & -\alpha_Z & \alpha_Y \\ \alpha_Z & 1 & -\alpha_X \\ -\alpha_Y & \alpha_X & 1 \end{bmatrix}. \quad (4.12)$$

Natomiast dla przesunięć o wektor  $T(t_X, t_Y, t_Z)$  macierz translacji ma postać:

$$\mathbf{T} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}. \quad (4.13)$$

Współrzędne punktu  $P$  na obrazie  $p(x, y)$  opisuje następująca zależność:

$$\begin{aligned} x &= \frac{fX}{Z}, \\ y &= \frac{fY}{Z}, \end{aligned} \quad (4.14)$$

gdzie  $f$  jest ogniskową kamery, zaś  $(X, Y, Z)$  są współrzędnymi punktu w przestrzeni.

### 4.3 Szacowanie przesunięcia endoskopu

Opracowany algorytm estymacji ruchu bronchoskopu rozwiązuje problem detekcji ruchu własnego, nie na podstawie punktów charakterystycznych, ale na podstawie ruchu ścian drzewa oskrzelowego (Twardowski i inni, 2006). Poprzez korelowanie obrazów ścian, po wcześniejszym usunięciu zniekształceń wprowadzanych przez optykę endoskopu, możliwe jest szybkie i dokładne wyznaczenie zarówno przesunięć końcówki endoskopu w głąb drzewa oskrzelowego, jak i wyznaczenie rotacji endoskopu. Duża wydajność obliczeniowa algorytmu jest konsekwencją przyjętego uproszczonego modelu segmentów drzewa oskrzelowego.

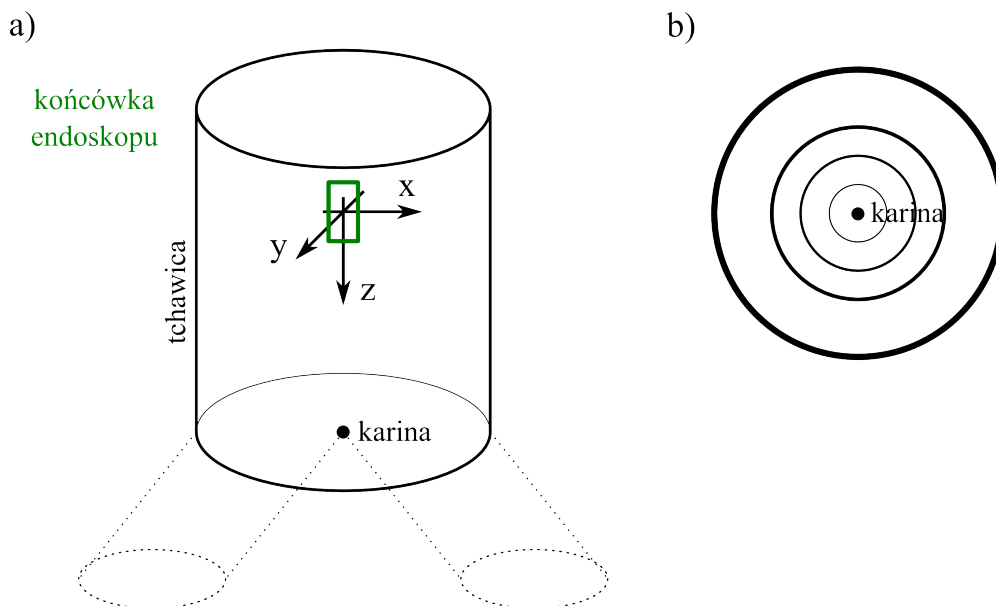
Głównym założeniem metody jest uproszczone przedstawienie tchawicy w postaci cylindra. Dodatkowym założeniem jest śledzenie rozbłysku światła na karinie (Daniilidis, 1997). Rozbłysk ten traktowany jest jako punkt odniesienia, na który patrzy kamera. Umożliwia on ograniczenie do czterech stopni swobody ruch kamery (przesunięcie przód-tył oraz trzy obroty). Ruch postępowy oraz obroty estymowane są na podstawie korelacji obrazu ścian obliczanej w cylindrycznym układzie współrzędnych.

Na rysunku 4.5a przedstawiono przyjęty model tchawicy, wraz z zaznaczoną pozycją końcówki endoskopu, przyjętym układem współrzędnych oraz śledzonym głównym rozwidleniem (kariną). Przyjęto, że współrzędne  $x$  oraz  $y$  wyznaczają płaszczyznę obrazu, zaś oś  $z$  skierowana jest w kierunku kariny. Obraz tchawicy powstający na sensorze przedstawiono na rysunku 4.5b.

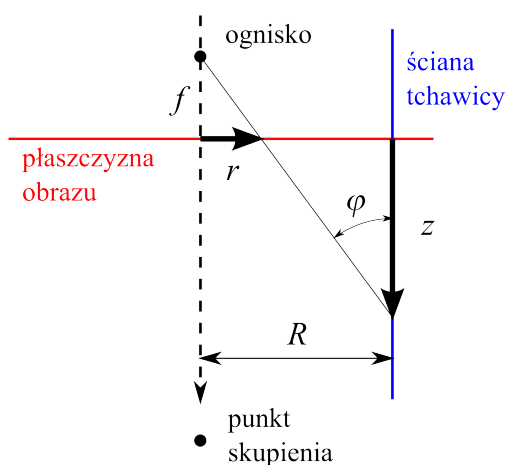
Otoczenie końcówki bronchofiberoskopu w drzewie oskrzelowym zostało zamodelowane w postaci cylindra. Ruch ścian drzewa oskrzelowego obserwowany na płaszczyźnie obrazu, odbywający się względem ustalonego punktu charakterystycznego, może być wyznaczony z wykorzystaniem korelacji obszarów kolejnych ramek sekwencji wideo po przejściu do układu biegunowego. Oczywiście tak znaczne uproszczenie opisu otoczenia prowadzi do znacznych błędów estymacji położenia endoskopu. Jednak ze względu na prostotę i szybkość, metoda ta może znaleźć zastosowanie w zgrubnym oszacowaniu ruchu endoskopu.

Zależności geometryczne w przyjętym uproszczonym modelu zostały przedstawione na rysunku 4.6. Zostały one wprowadzone dla współosiowego ustawienia kamery.





Rysunek 4.5: (a) Uproszczony model tchawicy przyjęty w szybkiej metodzie śledzenia oraz (b) obraz ścian modelu tchawicy widziany z pozycji końcówki endoskopu (Twardowski i inni, 2006)



Rysunek 4.6: Model powstawania obrazu w środowisku cylindrycznym.  $R$  - promień cylindra,  $f$  - długość ogniskowej,  $r$  - odległość punktu na obrazie,  $z$  - głębokość punktu na ścianie (Twardowski i inni, 2006)

Relacja pomiędzy odległością  $z$  oraz  $r$  może być wyprowadzona z następującej zależności geometrycznej:

$$\frac{R-r}{z} = \operatorname{tg} \varphi = \frac{R}{z+f}, \quad (4.15)$$

co po przekształceniach daje:

$$r = R \left( 1 - \frac{z}{z+f} \right). \quad (4.16)$$

Można zauważyć, że promień cylindra  $R$  jest wyłącznie czynnikiem skalującym.

Minimalna odległość  $z$  w polu widzenia kamery jest związana z kątem widzenia kamery  $\alpha$  i odległością od ściany  $R$ , która jest równa średnicy cylindra przy współosiowym ustawieniu

kamery:

$$z_{min} = \frac{R}{\operatorname{tg}(\alpha)} - f. \quad (4.17)$$

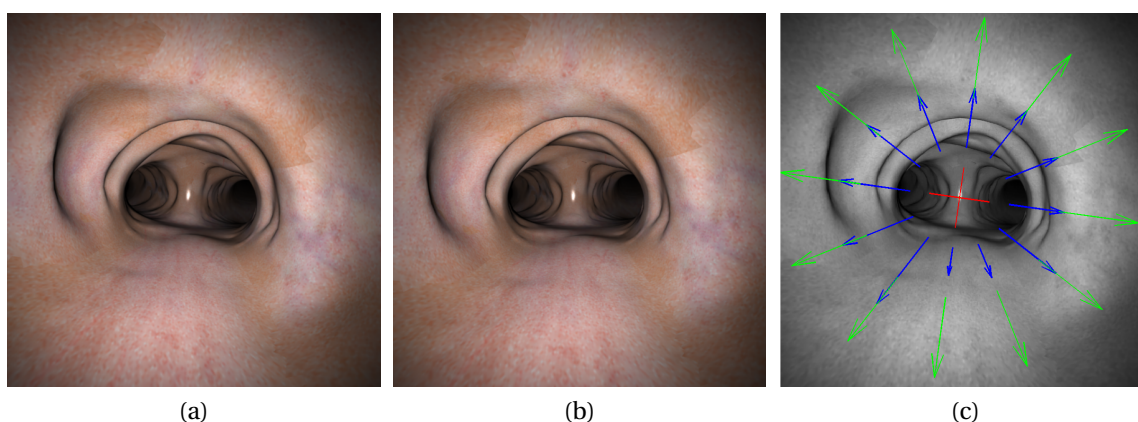
Analogicznie, odległość maksymalna wyznaczana jest z następującej zależności:

$$z_{max} = \frac{R}{\operatorname{tg}(\alpha_{min})} - f. \quad (4.18)$$

Świadomie przyjęte, w powyższym opisie matematycznym, uproszczenie dotyczy centralnego położenia kamery, a także centralnego położenia punktu, na który patrzy kamera (punktu skupienia lub fiksacji).

### 4.3.1 Wyznaczanie ruchu endoskopu

Zakładając uproszczony model dróg oddechowych, estymacja ruchu może być podzielona na trzy etapy: wyznaczenie rozbłysku na karinie (punktu fiksacji), oszacowanie kąta obrotu kamery oraz przesunięcia wzdłuż ścian drzewa oskrzelowego. Na rysunku 4.7 przedstawiono dwa obrazy z wirtualnej bronchoskopii (otrzymane z wykorzystaniem opracowanego modelu zabiegu bronchofiberoskopii) oraz wynik działania algorytmu estymacji ruchu endoskopu. Na zaprezentowanych obrazach wirtualna kamera została przesunięta w kierunku rozwidlenia oraz obrócona o kąt  $6^\circ$ . Otrzymane wektory ruchu zostały przeskalowane, w celu poprawienia czytelności rysunku. Opis działania algorytmu znajduje się w dalszej części rozdziału.



Rysunek 4.7: Wynik estymacji ruchu: (a-b) dwa obrazy wirtualnej bronchoskopii oraz (c) otrzymane cząstkowe wektory ruchu dla 12 fragmentów tchawicy (—), obliczony wektor ruchu (—) oraz znacznik obrotu (—)

Rozbłysk na karinie stanowi dobry punkt odniesienia dla transformacji pikseli obrazu do biegunowego układu współrzędnych. Śledzenie tego punktu nie jest kosztowne obliczeniowo, ponieważ opiera się na znalezieniu pikseli o maksymalnej jasności w ograniczonym obszarze. Zakładając małą dynamikę zmian kierunku patrzenia kamery endoskopu, śledzenie rozbłysku można zrealizować za pomocą przestrzennego okna przesuwanego wraz ze zmianą pozycji rozbłysku w kolejnych klatkach filmu. W pracy (Twardowski i inni, 2006) zaproponowano użycie okna Gaussa:

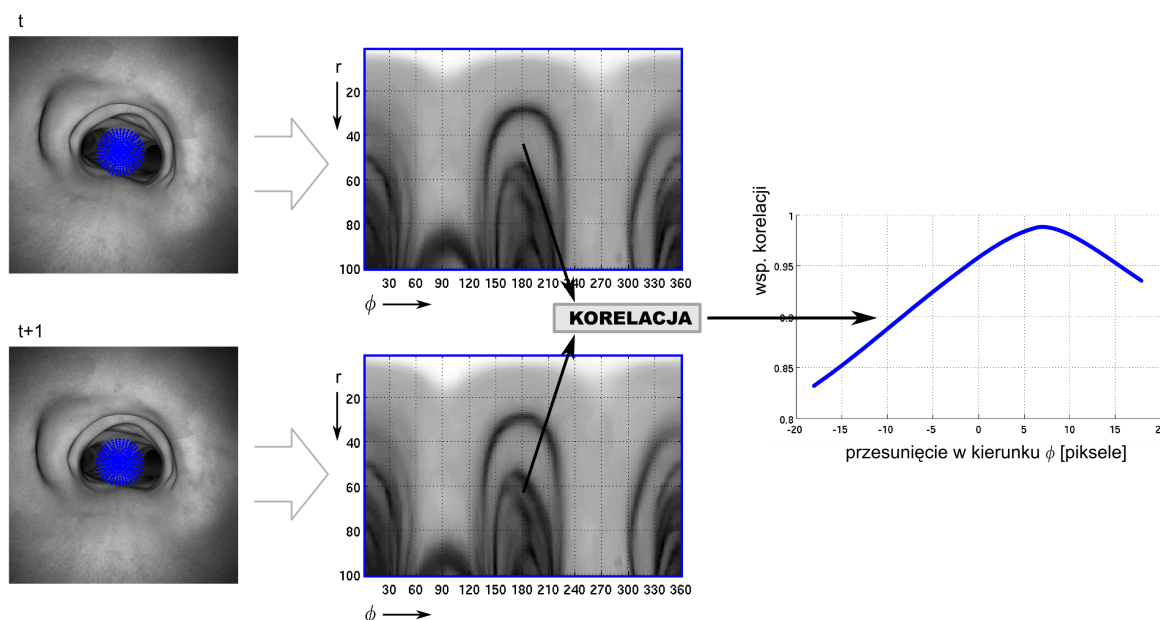
$$W(x, y) = \exp \left[ -\frac{1}{D^2} \left( (x - x_c)^2 + (y - y_c)^2 \right) \right], \quad (4.19)$$

gdzie  $D$  jest szerokością okna, zaś  $\{x_c, y_c\}$  to współrzędne pikselowe aktualnego rozbłysku. Kolejny punkt fiksacji wyznaczany jest natomiast z następującej zależności:

$$\{x_c, y_c\} = \max_{x,y} [I(x, y) \cdot W(x, y)], \quad (4.20)$$

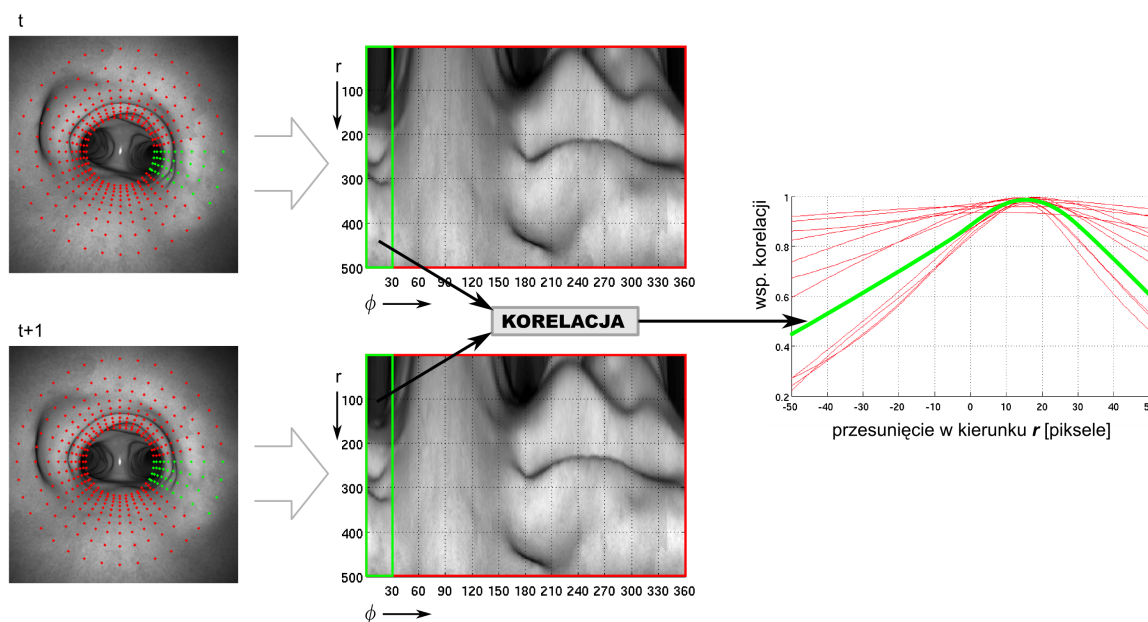
gdzie  $I$  jest obrazem z endoskopu, zaś  $W$  zastosowanym oknem. Z przeprowadzonych testów wynika, że tak prosta metoda śledzenia rozbłysku jest stabilna przy jednoczesnej dużej wydajności.

Kąt obrotu kamery wyznaczany jest na podstawie analizy otoczenia punktu fiksacji. Rysunek 4.8 przedstawia zasadę działania algorytmu estymacji obrotu endoskopu. Punkty obrazu należące do najbliższego otoczenia punktu fiksacji są transformowane do biegunowego układu współrzędnych o współrzędnych  $(r, \phi)$ , gdzie biegunem jest wyznaczony punkt rozbłysku. Obrót kamery endoskopu objawia się w takim układzie współrzędnych jako przesunięcie w osi  $\phi$ , które wykrywane jest poprzez korelowanie tak przygotowanych fragmentów kolejnych ramek obrazu.



Rysunek 4.8: Zasada działania algorytmu do szacowania obrotu endoskopu: dla obrazów z dwóch chwil czasowych, otoczenie rozbłysku zostaje przetransformowane do biegunowego układu współrzędnych, tak otrzymane obrazy są korelowane w kierunku osi  $\phi$  a uzyskana współrzędna maksimum współczynnika korelacji określa kąt obrotu

Przesunięcie kamery względem ścian drzewa oskrzelowego jest również wyznaczane poprzez korelację fragmentów oryginalnego obrazu po zmianie układu współrzędnych na biegunowy, jednak w tym przypadku korelowane są fragmenty znajdujące się dalej od punktu fiksacji. Zakres korelowanych obrazów zależy od minimalnej i maksymalnej odległości z rejestrowanej w polu widzenia kamery (równania (4.17) i (4.18)). Analizowany obszar ma więc kształt „obwarzanka”, który zostaje „rozwinęty” w wyniku przejścia do biegunowego układu współrzędnych. Zasada działania algorytmu została przedstawiona na rysunku 4.9. Współrzędne zaznaczonych punktów zostały wyliczone na podstawie przyjętego rurowego modelu otoczenia. Kolorem zielonym zaznaczono jeden wycinek, na podstawie którego wyznaczony jest wektor przesunięcia fragmentu ścian.



Rysunek 4.9: Zasada działania algorytmu szacowania przesunięcia endoskopu: dla obrazów z dwóch chwil czasowych, obraz ścian drzewa oskrzelowego zostaje przetransformowany do biegunowego układu współrzędnych  $(r, \phi)$ , tak otrzymane obrazy dzielony są na fragmenty (zielone ramki), które są korelowane ze sobą w celu określenia przesunięcia ścian

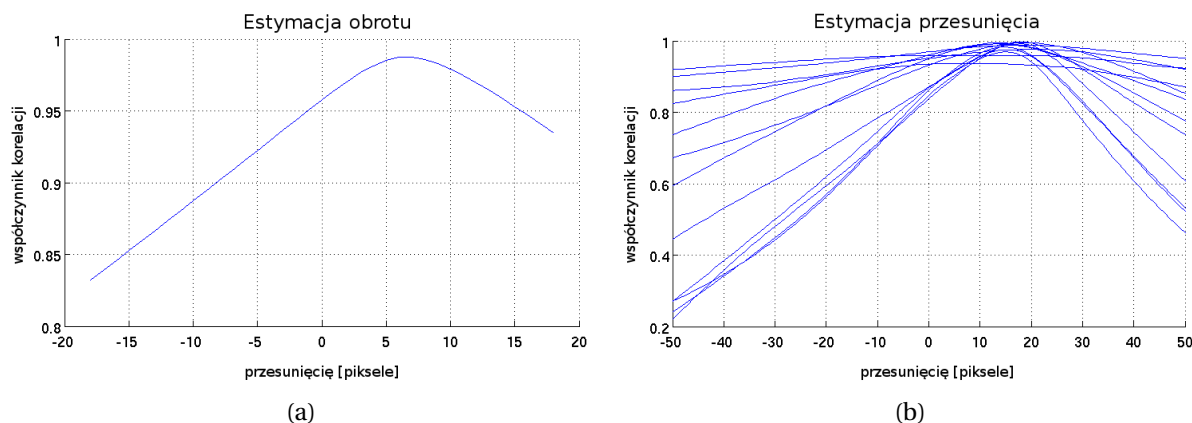
W przypadku ruchu postępowego kamery wzdłuż ścian tchawicy, ruch ten objawia się jako przesunięcie obrazu ścian w osi  $r$  biegunowego układu współrzędnych. W celu poprawy skuteczności działania algorytmu, „rozwinęty” obraz ścian drzewa oskrzelowego dzielony jest na  $k$  mniejszych fragmentów, które są następnie korelowane z odpowiadającymi im fragmentami obrazu poprzedniej ramki (rysunek 4.9). Podział obrazu na  $k$  fragmentów umożliwia lokalną analizę ruchu postępowego w zadanej liczbie kierunków. Matematyczny opis wyznaczania przesunięcia ma postać:

$$v_k = \max_v \text{corr} [I_{pn,1}(r_n, \phi_k), I_{pn,2}(r_n, \phi_k)], \quad (4.21)$$

gdzie:  $I_{pn}$  są składowymi rotacji i przesunięcia wyliczonymi dla fragmentów obrazu w układzie polarnym, po uwzględnieniu korekcji perspektywy, a  $\phi_k$  jest zakresem kątów w  $k$  kierunkach.

Wyznaczenie kąta obrotu polega na korelowaniu obrazów w osi  $\phi$ , zaś ruchu postępowego w osi  $r$  biegunowego układu współrzędnych. W obu przypadkach do obliczenia przesunięcia stosowana była unormowana wersja korelacji. Rysunek 4.10 przedstawia obliczone wartości unormowanej korelacji obrazów przedstawionych na rysunkach 4.8 i 4.9. W przypadku estymacji obrotu, maksymalna wartość współczynnika korelacji została wyznaczona dla przesunięcia 6 pikseli (rysunek 4.10a). Ponieważ rozmiar korelowanego fragmentu w osi  $\phi$  wynosił 360 pikseli, przesunięcie o 6 pikseli odpowiada obrotowi o  $6^\circ$ , co jest zgodne z wartością zadaną.

Zaprezentowane na rysunku 4.10b współczynniki unormowanej korelacji odpowiadają dwunastu fragmentom jednakowej wielkości, na które zostały podzielone obrazy z rysunku 4.9. Można zauważyć, że dla kolejnych fragmentów wyznaczone maksymalne wartości korelacji osiągane są dla różnych wartości przesunięcia. Oznacza to, że różne fragmenty obrazu ścian drzewa oskrzelowego przesuwały się z różną prędkością. Uzyskane, dla poszczególnych fragmentów obrazów, wektory ruchu przedstawiono na rysunku 4.7c. Ponieważ różnice między poszczególnymi obrazami są niewielkie, wektory ruchu byłyby niewidoczne na rysunku.



Rysunek 4.10: Współczynniki korelacji: (a) otrzymane przy wyznaczaniu obrotu (rysunek 4.8) oraz (b) otrzymane przy wyznaczaniu przesunięcia (rysunek 4.9)

Dlatego zostały one przeskalowane. Biorąc pod uwagę skomplikowany kształt drzewa oskrzelowego, przedstawionego na analizowanych obrazach wirtualnej bronchoskopii, wyznaczone różne przesunięcia fragmentów ścian można uznać za poprawne.

Ostatecznie, wektor przesunięcia końcówki endoskopu obliczany jest jako średnia geometryczna wektorów cząstkowych. Dodatkowo, średnia arytmetyczna wektorów cząstkowych ruchu umożliwia obliczenie przesunięcia endoskopu w płaszczyźnie obrazu, będącej skutkiem ruchu „w bok” lub ugięcia (obrotu) końcówki endoskopu.

### 4.3.2 Koszty obliczeniowe algorytmu

Analiza kosztów obliczeniowych algorytmu szacowania przesunięcia końcówki endoskopu została najpierw wykonana z wykorzystaniem środowiska Matlab, w którym powstała pierwsza implementacja tego algorytmu. Testy przeprowadzono na obrazach pochodzących z wideoendoskopu o rozdzielczości 369 na 382 pikseli. Do oszacowania udziału poszczególnych fragmentów rozbudowanego programu użyto funkcji *profile*, która umożliwia zliczenie ilości wywołań konkretnych funkcji oraz obliczenie czasu ich wykonywania. Testy przeprowadzono na komputerze z procesorem Core2Quad z zegarem 2,4GHz.

Algorytm opiera się głównie na porównywaniu obrazów pochodzących z dwóch różnych chwil czasowych. Miarą podobieństwa jest funkcja korelacji obrazów. Ten fragment kodu jest wykonywany bardzo często, potwierdzają to wyniki eksperymentu. Estymacja przesunięcia endoskopu obliczona dla dwóch obrazów trwała 5,248s, w tym czasie funkcja *corr2* służąca do wyliczenia współczynnika korelacji obrazów została wywołana 1249 razy, co trwało łącznie 2,120s. Przy czym 37 wywołań przypadło na estymację kąta rotacji, zaś 1212 na estymację przesunięcia. Można więc przyjąć, że około 40% czasu wykonywania algorytmu przypada na obliczanie funkcji korelacji.

Drugim istotnym fragmentem kodu jest przygotowanie danych do korelacji, które polega na transformowaniu fragmentów obrazów do biegunowych układów współrzędnych. Operacje te wiążą się z koniecznością dokonania interpolacji danych. Przygotowanie danych do estymacji obrotu trwało około 0,417s, zaś samej estymacji obrotu 0,591s. W tym czasie dwukrotnie wygenerowano obrazy o rozdzielczości odpowiednio 360 na 100 pikseli oraz 500 na 360 pikseli.

W obydwu przypadkach wywoływana była funkcja *interp2*, która interpolowała dane w czasie 0,799s, co stanowi ponad 15% czasu działania algorytmu.

Wyznaczanie współrzędnych punktu rozbłysku, zgodnie z przewidywaniami, było szybkie i stanowiło jedynie 3% udziału w całkowitych kosztach obliczeniowych.

Podsumowując można stwierdzić, że istnieją dwa kluczowe fragmenty kodu algorytmu szacowania położenia endoskopu, których optymalizacja może wpłynąć na skrócenie czasu obliczeń. Pierwszym jest funkcja korelacji, która jest wywoływana bardzo często i na którą przypada około 40% czasu pracy algorytmu zaimplementowanego w środowisku Matlab. Drugim fragmentem jest interpolacja danych, a ściślej generowanie obrazów do korelacji, na który przypada około 15% czasu obliczeń.

### 4.3.3 Korelacja

Funkcja korelacji jest wartością oczekiwaną iloczynu zmiennych losowych, która dla sygnałów niestacjonarnych jest zależna od wyboru wartości chwili czasowej  $t$  oraz przesunięcia  $\tau$  (Zieliński, 2007):

$$R_{xy}(t, \tau) = E[x(t)y(t-\tau)] = \iint_{-\infty}^{+\infty} x_t y_{t-\tau} p(x_t y_{t-\tau}) dx_t dy_{t-\tau}, \quad (4.22)$$

zaś dla sygnałów stacjonarnych, zależy jedynie od przesunięcia:

$$R_{xy}(\tau) = E[x(t)y(t-\tau)] = \iint_{-\infty}^{+\infty} x_t y_{t-\tau} p(x_t y_{t-\tau}) dx_t dy_{t-\tau}. \quad (4.23)$$

Na potrzeby przetwarzania sygnałów współczynnik korelacji definiowany jest w następujący sposób:

$$R(t) = f(t) \circ g(t) = \int_{-\infty}^{+\infty} f(t) g^*(t-\tau) d\tau. \quad (4.24)$$

gdzie  $f^*(\tau)$  jest sprzężoną wartością zespolonej funkcji  $f(x)$ .

Funkcja korelacji dwóch sygnałów, znajduje szczególne zastosowanie jako miara ich podobieństwa. W przypadku dwuwymiarowych sygnałów dyskretnych, jakimi są obrazy cyfrowe, funkcja *korelacji wzajemnej* (ang. *cross-correlation*) może być wyznaczona z następującej zależności:

$$C(i, j) = \sum_{m=0}^{(M_a-1)} \sum_{n=0}^{(N_a-1)} A(m, n) \cdot B^*(m+i, n+j), \quad (4.25)$$

gdzie:  $A$  jest obrazem o rozmiarach  $(M_a, N_a)$ ,  $B$  jest obrazem o rozmiarach  $(M_b, N_b)$ , zaś indeksy przesunięć  $(i, j)$  spełniają następujące zależności:  $0 \leq i < (M_a + M_b - 1)$  oraz  $0 \leq j < (N_a + N_b - 1)$ .

**Korelacja wzajemna obrazów obliczana z wykorzystaniem transformaty Fouriera.** Algorytm korelacji obliczany „z definicji” jest mało efektywny, a jego złożoność obliczeniowa rośnie z kwadratem liczby korelowanych danych, czyli jest klasy  $n^2$ . Jedną z częściej stosowanych metod zmniejszania kosztów obliczeniowych jest zastosowanie metody bazującej na trzech szybkich transformacjach Fouriera (ang. *Fast Fourier Transformation* - FFT).

W przypadku sygnałów dyskretnych szybki algorytm wyznaczania funkcji korelacji sygnałów składa się z następującej sekwencji obliczeń (Zieliński, 2007):

1. wyznaczenie transformat Fouriera korelowanych sygnałów:

$$X(j\omega) = \mathbf{F}\{x(t)\}; \quad Y(j\omega) = \mathbf{F}\{y(t)\}, \quad (4.26)$$



2. obliczenie iloczynu transformat będących wartościami funkcji korelacji w dziedzinie częstotliwości:

$$R(j\omega) = X(j\omega) \cdot Y^*(j\omega), \quad (4.27)$$

3. wyznaczenie wartości funkcji korelacji w dziedzinie czasu poprzez odwrotną transformatę Fouriera

$$X(t) = \mathbf{F}^{-1} \{R(j\omega)\}. \quad (4.28)$$

Przy użyciu szybkiej wersji transformaty Fouriera algorytm ten ma złożoność obliczeniową klasy  $n \log(n)$  i jest szczególnie przydatny przy korelowaniu dużych zbiorów danych. W przypadku mniejszych zbiorów danych, koszty uruchomienia algorytmów szybkiej transformacji Fouriera oraz zarządzania pamięcią znacznie zmniejszają teoretyczne możliwości przyspieszenia obliczeń.

**Unormowana korelacja wzajemna obrazów.** Innym algorytmem obliczania podobieństwa obrazów jest algorytm *unormowanej korelacji wzajemnej* (ang. *normalized cross correlation* — *NCC*) (Lewis, 1995; Briechle i Hanebeck, 2001; Hii i inni, 2006):

$$NCC_{I,T}(u, v) = \frac{\sum_{x,y} (I(x, y) - \bar{I}) \cdot (T(x - u, y - v) - \bar{T})}{\sqrt{\sum_{x,y} (I(x, y) - \bar{I})^2 \cdot \sum_{x,y} (T(x - u, y - v) - \bar{T})^2}}, \quad (4.29)$$

gdzie  $I$  jest obrazem o wymiarach  $(M_x, M_y)$ ,  $T$  jest szukanym wzorcem o rozmiarach  $(N_x, N_y)$ ,  $\bar{I}$  jest wartością średnią wzorca,  $(u, v)$  jest przesunięciem, dla którego obliczana jest korelacja, zaś  $\bar{I}$  jest wyznaczane z zależności:

$$\bar{I} = \frac{1}{N_x N_y} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} I(x, y). \quad (4.30)$$

Algorytm unormowanej korelacji wzajemnej obrazów znalazł przede wszystkim zastosowanie w wyszukiwaniu i śledzeniu znaczników w obrazach wideo. Umożliwia on wyznaczenie pozycji niewielkiego fragmentu obrazu (wzorca) w większym otoczeniu. Zaproponowana w pracy (Lewis, 1995) metoda przyspieszenia obliczania NCC opiera się na wyeliminowaniu kosztownych obliczeniowo normalizacji (mianownik równania (4.29)) poprzez zastosowanie tablic sum cząstkowych (ang. *running sum*). W pracy (Lewis, 1995) przedstawiono szczegółowe wyprowadzenie. Natomiast współczynnik korelacji (licznik równania (4.29)) obliczany jest z wykorzystaniem transformaty Fouriera.

Kolejnym sposobem przyspieszenia obliczania licznika jest zastąpienie wzorca sumą prostych funkcji bazowych (Briechle i Hanebeck, 2001). W przypadku korelowania dużych zbiorów danych, jak ma to miejsce w opisanym w metodzie szacowania przesunięcia endoskopu, przyspieszona metoda NCC byłaby trudna w zastosowaniu — cały jeden korelowany obraz musiałby być zamieniony na obraz uproszczony.

## 4.4 Dokładne dopasowywanie obrazów metodą informacji wzajemnej

Wynikiem szybkiej metody estymacji położenia endoskopu, opisanym w rozdziale 4.3, jest zgrubne oszacowanie pozycji endoskopu. W celu wyznaczenia dokładnej pozycji kamery wykorzystano dopasowanie obrazów rzeczywistego oraz wirtualnego (Higgins i inni, 2003; Mori



i inni, 2004). Algorytmy dopasowywania obrazów są zazwyczaj iteracyjne i składają się z następujących zadań:

1. obliczenia transformacji geometrycznej obrazu, będącej wynikiem ruchu kamery i obejmującej macierze: rotacji, translacji i skalowania,
2. wyznaczenia stopnia podobieństwa dopasowywanych obrazów,
3. dobrania optymalnych parametrów transformacji obrazów, zapewniającej największe podobieństwo porównywanych obrazów.

Ponieważ porównywane miały być obrazy z różnych źródeł, posiadające odmienny charakter, do implementacji wybrano miarę podobieństwa, bazującą na entropii obrazów – informacji wzajemnej obrazów będącej miarą rozproszenia histogramu wzajemnego (Higgins i inni, 2003; Turcza i Duplaga, 2004). W celu ograniczenia liczby generowanych obrazów wirtualnych w kolejnych pracach zaproponowano iteracyjny, gradientowy algorytm optymalizacji oraz dopasowywanie w schemacie wielorozdzielczym (Turcza, 2005b,a).

Miara podobieństwa obrazów bazująca na teorii informacji została zaproponowana w pracy (Viola i Wells, 1995). Opiera się ona na koncepcji wspólnej entropii podanej przez Shannona:

$$M(u, v) = H(u) - H(u|v) \quad (4.31)$$

gdzie:  $M(u, v)$  jest *informacją wzajemną* — miarą zależności pomiędzy dwoma zmiennymi losowymi,  $H(u)$  jest miarą niepewności co do wartości zmiennej losowej  $u$  (entropia),  $H(u|v)$  jest miarą niepewności co do wartości zmiennej losowej  $u$ , ustalonej przy założeniu, że wartość zmiennej losowej  $v$  jest znana (entropia warunkowa). Przy takim zapisie, informacja wzajemna  $M(u, v)$  wyraża jak niepewność wartości  $u$  zmniejsza się po poznaniu wartości  $v$ . Jeżeli wartość entropii warunkowej  $H(u|v)$  maleje, wartość informacji wzajemnej  $M(u, v)$  rośnie. Korzystając z twierdzenia Bayesa, wyrażające prawdopodobieństwo warunkowe zdarzeń  $A$  i  $B$ :

$$P(A, B) = P(A|B)P(B) \quad (4.32)$$

oraz zapisując entropie z równania (4.31) w postaci:

$$H(u) = - \sum_i p_u(i) \log p_u(i), \quad (4.33)$$

$$H(u, v) = - \sum_{i,j} p_{uv}(i, j) \log p_{uv}(i, j), \quad (4.34)$$

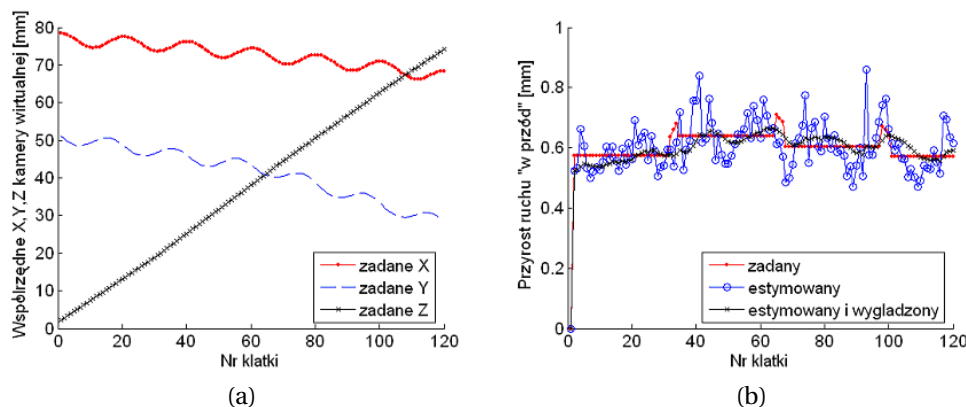
ostatecznie, równanie na informację wzajemną może być zapisane w następujący sposób:

$$I(u, v) = H(u) + H(v) - H(u, v). \quad (4.35)$$

W równaniu 4.35 występuje entropia wspólna  $H(u, v)$ , która może być określana na podstawie wspólnego rozkładu prawdopodobieństwa. Rozkład ten może być wyznaczony na podstawie *histogramu wzajemnego*  $h(u, v)$  po jego normalizacji (Higgins i inni, 2003; Turcza, 2005a).

## 4.5 Weryfikacja metody

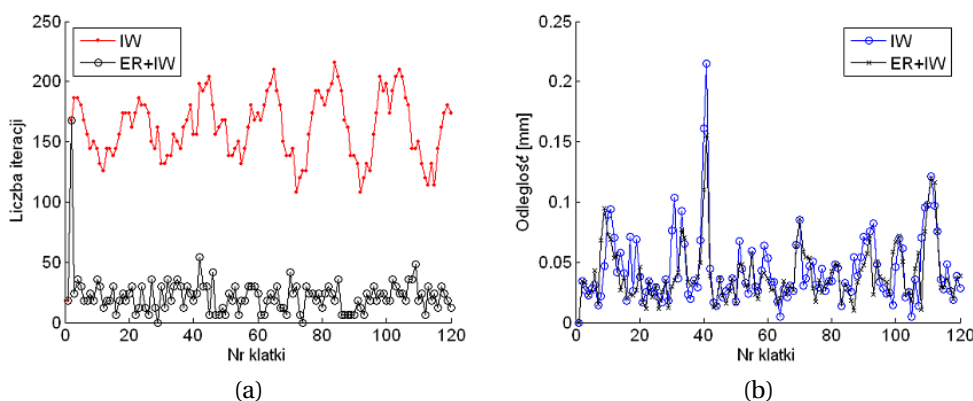
Na rysunku 4.11 przedstawiono wyniki eksperymentu weryfikującego poprawność działania algorytmu śledzenia endoskopu, w którym wygenerowano dwie ścieżki ruchu wirtualnej kamery, znajdującej się w drzewie oskrzelowym, a następnie wykorzystano je jako obrazy z rzeczywistego endoskopu. Celem eksperymentu było wyznaczenie błędu estymacji przesunięcia wzdłuż osi drzewa oskrzelowego (ruch „w przód“ i „w tył“).



Rysunek 4.11: Wyniki testów na obrazach syntetycznych: (a) współrzędne ( $X, Y, Z$ ) trajektorii ruchu kamery „po śrubie”, (b) estymowane przesunięcie endoskopu w kierunku kariny (Zieliński i inni, 2007)

Na rysunku 4.11a przedstawiono współrzędne trajektorii, po której poruszała się kamera. Był to ruch jednostajny w kierunku kariny z jednoczesnym ruchem po elipsie w płaszczyźnie obrazu (ruch „po śrubie”). Ruch w płaszczyźnie obrazu symulował niewspółosiowe przemieszczanie się endoskopu w drzewie oskrzelowym. Wynik estymacji ruchu postępowego przedstawiono na rysunku 4.11b. Można zauważyć, że uzyskany błąd estymacji dla dość skomplikowanego ruchu nie przekracza wartości  $0,2\text{ mm}$ . Wprowadzenie dodatkowego wygładzania znacznie ogranicza fluktuacje estymowanego przesunięcia.

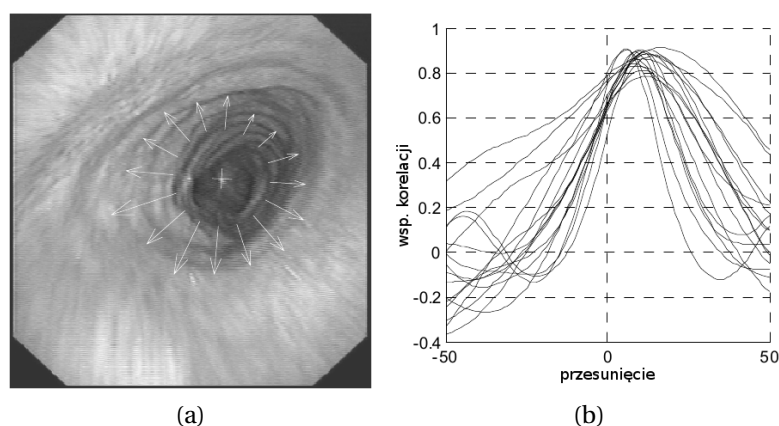
Zbadano również liczbę dopasowań obrazów przeprowadzonych w trakcie estymacji ruchu wirtualnej kamery. Wyniki zaprezentowano na rysunku 4.12, który przedstawia liczbę iteracji algorytmu w funkcji numeru klatki symulowanej wirtualnej bronchoskopii (ruch postępowy „po śrubie“, rysunek 4.11a).



Rysunek 4.12: Wyniki testów algorytmu estymacji położenia końcówki endoskopu na obrazach wirtualnych: (a) liczba iteracji pętli dopasowania obrazów przy wykorzystaniu jedynie informacji wzajemnej (IW) oraz w połączeniu z szybką estymacją ruchu (ER+IW), (b) obliczona odległość euklidesowa pomiędzy zadaną a wyznaczoną przez algorytm pozycją kamery (Zieliński i inni, 2007)

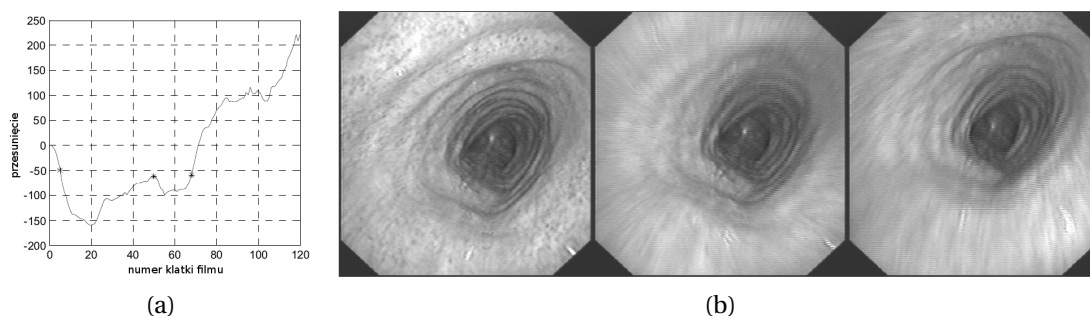
Można zauważyć, że wyznaczenie pozycji wirtualnej kamery jedynie z wykorzystaniem informacji wzajemnej (IW) wymaga od 130 do 220 iteracji. Po zastosowaniu wstępnej, szybkiej estymacji ruchu (ER) liczba iteracji koniecznych do wyznaczenia pozycji endoskopu spada do poziomu około 25. Dodatkowo, redukcja liczby iteracji nie wiąże się ze zwiększeniem błędu estymacji, co zaprezentowano na rysunku 4.12b. Przedstawia on odległość euklidesową obliczoną dla każdej klatki testowej wirtualnej bronchoskopii, obliczonej między zadanymi współrzędnymi wirtualnej kamery a współrzędnymi wyznaczonymi przez algorytm.

Na rysunku 4.13 przedstawiono przykładowy obraz z endoskopu, dla którego metodą zgrubną wyznaczono wektory ruchu. Na ich podstawie szacowane jest przesunięcie endoskopu.



Rysunek 4.13: Wynik testów algorytmu estymacji ruchu na obrazach rzeczywistych: (a) przykład klatki z zaznaczonymi wektorami ruchu obliczonymi szybką metodą szacowania przesunięcia (rozdział 4.3), (b) wartości funkcji korelacji obliczone dla poszczególnych fragmentów obrazu (Twardowski i inni, 2006)

Kolejny rysunek 4.14 przedstawia obliczone dla 120 klatek filmu przesunięcie endoskopu. Można zauważyć, że w początkowej fazie ruchu endoskop został wycofany z tchawicy, a następnie wsunięty na większą głębokość. Na wykresie przesunięcia zaznaczono trzy klatki filmu, przedstawione obok wykresu, które prezentują ten sam fragment drzewa oskrzelowego.



Rysunek 4.14: Wynik testów algorytmu estymacji ruchu na obrazie rzeczywistym: (a) skumulowane przesunięcie wzdłuż tchawicy, na którym zaznaczono gwiazdką (\*) trzy klatki filmu o numerach 5, 50 i 68 (b), znajdujące się w podobnej lokalizacji (Twardowski i inni, 2006)

W pracy (Zieliński i inni, 2007) zaprezentowano wyniki przeprowadzonych testów wydajności opracowanego systemu. Testy wydajności miały na celu oszacowanie złożoności obliczeniowej poszczególnych bloków systemu nawigacyjnego, wykrycie najwolniejszych jego fragmentów, a następnie zaproponowanie rozwiązań, które mogłyby zagwarantować pracę systemu w czasie rzeczywistym (analizę wszystkich obrazów docierających z endoskopu). Kamera endoskopu dostarcza 25 obrazów na sekundę. Prędkość napływu kolejnych obrazów z endoskopu determinuje najdłuższy czas (równy 40ms), w którym niezbędne jest wyznaczenie nowej pozycji endoskopu poprzez przeprowadzenie dopasowania obrazu wirtualnego do rzeczywistego.

Przybliżone czasy wykonywania pojedynczych fragmentów kodu, opracowanego systemu nawigacji, przytoczono w tabeli 4.3. Czasy obliczeń zostały zmierzone w środowisku Matlab, zaś czas implementacji w C++ został oszacowany przy założeniu dziesięciokrotnego przyśpieszenia obliczeń po przejściu do języka niskiego poziomu. Założenie dziesięciokrotnego przyśpieszenia zostało poparte implementacją wybranych fragmentów kodu (Zieliński i inni, 2007).

Tablica 4.1: Czas wykonywania kluczowych fragmentów kodu dla opracowanego systemu uruchomionego na procesorze Pentium 4HT 3,2GHz (Zieliński i inni, 2007)

Operacja	Czas obliczeń [ms]	
	Matlab	C++
Korekcja nieliniowości obrazu	13	1,4
Estymacja ruchu endoskopu	4600	460
Generacja obrazu wirtualnego	—	1,4
Wyznaczenie informacji wzajemnej	46	4,9

Biorąc pod uwagę otrzymane czasy obliczeń (tabela 4.3) oraz uzyskane w wyniku eksperymentu na danych rzeczywistych liczby iteracji algorytmu (rysunek 4.12a) możliwe było oszacowanie czasu niezbędnego do obliczenia pozycji endoskopu. Szacunki te przytoczono w tabeli 4.2.

Tablica 4.2: Szacowany czas wyznaczenia pozycji endoskopu: wykonanie dopasowania obrazu wirtualnego do obrazu z endoskopu, dla systemu uruchomionego na procesorze Pentium 4HT 3,2GHz (Zieliński i inni, 2007)

Operacja	Czas obliczeń [ms]
Korekcja nieliniowości obrazu	$1 \times 1,4 = 1,4$
Estymacja ruchu endoskopu	$1 \times 460 = 460,0$
Generacja obrazu wirtualnego	$25 \times 1,4 = 35,0$
Wyznaczenie informacji wzajemnej	$25 \times 4,9 = 122,5$
Razem	618,9

Można zauważyć, że najbardziej czasochłonnym fragmentem systemu jest wstępna estymacja parametrów ruchu endoskopu, którego udział wynosi około 74% szacowanego czasu obliczeń. Pomimo znacznych kosztów fragment ten jest niezbędny, ponieważ przede wszystkim zabezpiecza algorytm optymalizacji używający informacji wzajemnej przed zatrzymaniem się w maksimach lokalnych. Dodatkowo redukuje około sześciokrotnie liczbę niezbędnych operacji dopasowywania obrazów metodą informacji wzajemnej (patrz rysunek 4.12).

Otrzymany sumaryczny czas estymowania pozycji endoskopu, równy 620ms, przekracza około 15 razy czas między kolejnymi obrazami z wideobronchoskopu. W pracy (Zieliński i inni, 2007) zaproponowano kilka możliwości szybkiej implementacji opracowanego modelu systemu do nawigacji. Wymieniono między innymi, możliwość implementacji z wykorzystaniem specjalizowanych i optymalizowanych bibliotek do przetwarzania danych, wykorzystanie zasobów sprzętowych zaawansowanych kart do akwizycji obrazów (np. firmy Matrox), równoległego (wielowątkowego) przetwarzania danych czy też strumieniowego przetwarzania danych (Wiatr, 2003).

Autor rozprawy zaimplementował najbardziej kosztowny fragment algorytmu zgrubnej estymacji ruchu końcówki endoskopu z wykorzystaniem biblioteki CUDA (NVIDIA, 2011b,a), która umożliwia wykonywanie obliczeń przy użyciu procesora GPU (ang. *Graphics Processing Unit*) karty graficznej.

## 4.6 Akceleracja obliczeń w technologii CUDA

Metoda szacowania położenia końcówki endoskopu opisana w rozdziale 4.3 została zaimplementowana w języku C++. Kolejnym krokiem było implementacja kluczowych fragmentów z wykorzystaniem biblioteki CUDA, która umożliwia wykonywanie obliczeń na procesorze karty graficznej (Socha, 2013). Użycie procesora graficznego GPU (ang. *Graphics Processing Unit*) do wspomaganie obliczeń jest obecnie szczególnie uzasadnione, biorąc pod uwagę powszechność występowania tych procesorów w nowoczesnych komputerach klasy PC oraz dostępność narzędzi programistycznych, ułatwiających wykonywanie dowolnych obliczeń z użyciem GPU. Wydajność nowoczesnego procesora GPU znacznie przekracza wydajność procesora CPU. Opis podstawowych możliwości i własności biblioteki CUDA znajduje się w dodatku A.2.

Programy uruchamiane w procesorze GPU realizują równoległe przetwarzanie danych typu SPMD (ang. *Single-Program Multiple-Data*). Polega ono na uruchomieniu na wielu procesorach tego samego programu (ang. *kernel*), wykonującego obliczenia na dużym zbiorze danych. Opracowane programy realizują równoległe przetwarzanie pikseli obrazu. Zaimplementowano dwa kluczowe pod względem czasu obliczeń fragmenty kodu: interpolację danych w celu przygotowania ich do dalszego przetwarzania oraz obliczanie unormowanej funkcji korelacji.

### 4.6.1 Przygotowanie danych

Na przygotowanie danych do korelacji składają się:

- przesłanie kolorowego obrazu do karty graficznej,
- korekcja zniekształceń,
- konwersja obrazu z kolorowego do odcieni szarości,
- wyznaczenie pozycji rozbłysku,
- interpolacja otoczenia rozbłysku,
- rozwinięcie ścian drzewa tchawicy.

Podstawowym założeniem podczas implementacji było przeniesienie wszystkich danych do pamięci karty graficznej, a następnie równoległe przetwarzanie ich z wykorzystaniem setek procesorów. W przypadku interpolacji danych, jeden piksel obrazu wynikowego obliczany jest przez jeden rdzeń GPU.

**Interpolacja danych.** Procesor GPU zawiera wyspecjalizowane jednostki do sprzętowego adresowania i filtrowania tekstur (rysunek A.8). Wykorzystanie tych jednostek umożliwia wydajne interpolowanie obrazów. Jednostki te mogą operować jedynie na specjalnym obszarze pamięci jakim jest pamięć tekstur (ang. *texture memory*). W pamięci tej umieszczane były obrazy, które wymagały adresowania subpikselowego: oryginalny obraz z endoskopu oraz obraz po korekcji zniekształceń i usunięciu koloru.

Korekcja zniekształceń obliczana była dla każdej klatki filmu pochodzącego z endoskopu. Obrazy wirtualnej bronchoskopii nie wymagały korekcji. Metodologię wyznaczania współczynników zniekształcenia obrazu opisano w rozdziale 3.3.2. Zależność między współrzędnymi pikseli obrazu źródłowego i skorygowanego opisuje równanie wielomianowe (3.39) (strona 52). Ponieważ przekształcenie to jest stałe dla konkretnego bronchoskopu, odpowiadające sobie współrzędne pikseli zostały stabelaryzowane (metoda *look-up table*). W tym celu utworzono dwuwymiarową tablicę o wymiarach obrazu po korekcji, w której umieszczono obliczone współrzędne pikseli obrazu nieskorygowanego. Tak przygotowaną tablicę umieszczono w obszarze pamięci tekstur (ang. *texture memory*).

W podobny sposób przygotowano tablice współrzędnych pikseli do estymacji kąta rotacji (rozwińcie okolic rozbłysku do polarnego układu współrzędnych) oraz przesunięcia (rozwińcie ścian drzewa oskrzelowego na cylinder). W obu przypadkach rozwińcie odbywa się wokół punktu fiksacji, którego współrzędne wyznaczone są dla każdej klatki na podstawie pozycji rozbłysku. Wartości stabelaryzowane wyznaczono dla punktu rozwińcia (początku biegunowego układu współrzędnych) o współrzędnych (0, 0). W trakcie tworzenia obrazów, współrzędne zapisane w tablicach są przesuwane o wartość równą współrzędnym bieżącego punktu fiksacji.

Program korygujący zniekształcenia wywoływany jest dla każdego piksela — rozmiar tablicy rdzeni (tzw. *Grid*) jest równy rozmiarom obrazu po korekcji. Każdy rdzeń, do którego trafia program, oblicza na podstawie zmiennych wbudowanych (patrz tabela A.2 na stronie 108) współrzędne  $(x_{out}, y_{out})$  „swojego” piksela. Współrzędne te służą do zaadresowania tablicy ze stabelaryzowanymi współrzędnymi pikseli  $(x_{in}, y_{in})$  obrazu źródłowego. W kolejnym kroku program, z wykorzystaniem funkcji *tex2D*, pobiera z obrazu źródłowego wartość koloru dla zadanych współrzędnych. Jeżeli zadane współrzędne nie są liczbami całkowitymi (nie „trafiają” w piksel), wykonywana jest dwuliniowa interpolacja danych. Ostatnim etapem algorytmu jest konwersja kolorowego obrazu do skali szarości poprzez obliczenie luminancji. Wynik zapisywany jest w tablicy znajdującej się w obszarze pamięci globalnej (ang. *global memory*), która zapewnia pełen dostęp do danych.

**Wyznaczanie współrzędnych rozbłysku.** Wyznaczenie współrzędnych rozbłysku odbywa się dwuetapowo. W pierwszej fazie monochromatyczny obraz pozbawiony zniekształceń wymnażany jest z wcześniej przygotowanym i stabelaryzowanym oknem Gaussa (wzór 4.19). Wartość maksymalna okna znajduje się w środku macierzy. Rozmiar tablicy zawierającej współczynnik okna jest dwa razy większy od rozmiaru obrazu po korekcji. Uprościło to wyznaczanie współrzędnych środka okna i rozwiązało problem sprawdzania zakresów tablic przy wymnażaniu współczynników okna z wartościami obrazu. Dodatkowo, możliwe jest wymnożenie okna z obrazem przy dowolnym przesunięciu.

Drugim krokiem było wyznaczenie współrzędnych rozbłysku występującego na rozwidleniu głównym tchawicy (wzór 4.20). Wydajna implementacja wyszukiwania wartości maksymalnej w środowisku wielordzeniowym nie jest zadaniem trywialnym, ponieważ wymaga przeglądnięcia i uwzględnienia wartości z wszystkich analizowanych danych. Operacja taka nie może

być zrealizowana jednocześnie przez wiele pracujących równolegle procesorów. Problem ten nosi nazwę *równoległej redukcji*, a jego rozwiązanie zostało szczegółowo przedstawione na przykładzie sumowania w pracy (Harris, 2008).

Sekwencyjne przeglądanie tablicy w poszukiwaniu wartości największej jest możliwe, ale zupełnie nieefektywne obliczeniowo. Wersja realizująca wyszukiwanie wartości maksymalnej opiera się na kaskadowym wywoływaniu programów: każdy rdzeń dokonuje porównania małego fragmentu danych, a wynik zapisywany jest do szybkiej pamięci wspólnej (ang. *shared memory*), tworząc dane wejściowe dla kolejnego uruchomienia programu. W kolejnych iteracjach algorytmu liczba porównywanych danych jest zmniejszana, aż ostatecznie otrzymywana jest jedna wartość wynikowa. Jak pokazują przykłady zaprezentowane w pracy (Harris, 2008), uwzględnienie architektury procesora GPU oraz optymalizacja kodu umożliwiają nawet trzydziestokrotne skrócenie czasu obliczeń.

Opracowana przez autora implementacja wyszukiwania współrzędnych rozbłysku bazuje na najszybszym przykładzie zaprezentowanym w pracy (Harris, 2008). Przykład ten został zmodyfikowany w taki sposób, by w wyniku obliczeń uzyskiwane były współrzędne rozbłysku.

**Przygotowanie macierzy do korelacji.** Po skorygowaniu zniekształceń, konwersji barw i wyznaczeniu punktu fiksacji (rozbłysku) uruchamiane są dwa programy, których zadaniem jest „rozwiniecie” odpowiednich fragmentów obrazu do układu biegunowego oraz na powierzchnię cylindra. Tak jak przy korygowaniu zniekształceń, również tutaj, obraz źródłowy (skorygowany i monochromatyczny) kopiowany jest z pamięci współdzielonej do pamięci tekstur. Dzięki temu, możliwe jest użycie funkcji *tex2D* do próbkowania ze sprzętową interpolacją pikseli obrazu źródłowego. Analogicznie, do programu korekcji zniekształceń współrzędne próbkowanych pikseli są odczytywane z tabeli. W kolejnym kroku tak otrzymane wartości są korygowane o offset wynikający z pozycji rozbłysku.

W wyniku działania programów dla jednej ramki obrazu wyznaczane są dwie macierze o zadanych rozmiarach, w których znajdują się „rozwiniete” okolice rozbłysku (rysunek 4.8) oraz ściany tchawicy (rysunek 4.9). Ponieważ obydwie zbiory danych korelowane są jednowymiarowo, ale w dwóch różnych kierunkach, macierz do wyznaczania przesunięcia jest transponowana już na poziomie tworzenia. Ma to na celu uproszczenie implementacji funkcji korelującej.

## 4.6.2 Korelacja obrazów

Przygotowane w pamięci karty graficznej tablice, zawierające przetransformowane fragmenty obrazów wejściowych, są korelowane z wykorzystaniem metody unormowanej korelacji wzajemnej NCC (strona 75) (Lewis, 1995).

Zaimplementowane obliczenia NCC prowadzone są w trzech etapach:

- w pierwszym kroku z obrazów usuwana jest wartość średnia: suma obrazów wyznaczana jest kaskadowo (równoległe, przez wiele procesorów) i dzielona przez rozmiar korelowanych danych,
- w kolejnym kroku obliczany jest czynnik normalizujący (mianownik równania (4.29)):
  - wartości pikseli są podnoszone do kwadratu, a następnie sumowane kaskadowo,
  - tak uzyskane sumy obrazów są wymnażane i pierwiastkowane,
- następnie wyznaczana jest wartość korelacji dla danego zbioru przesunięć w jednej osi, dla każdego przesunięcia  $u$  wykonywane są następujące obliczenia:



- dla każdego piksela fragmentu korelowanego obrazu uruchamiany jest program CUDA, którego zadaniem jest wymnożenie obrazu o współrzędnych  $I_1(x, y)$  z obrazem  $I_2(x - u, y)$ , w wyniku równoległego mnożenia obrazów powstaje macierz o rozmiarze takim samym jak korelowane obrazy,
- otrzymana macierz zostaje zsumowana (licznik równania (4.29)),
- wynik sumowania jest normowany wcześniej obliczoną wartością i zapisywany do macierzy wyników, pod indeksem  $u$ .

W kolejnych iteracjach programu obliczenia wykonywane na najnowszym obrazie są zapisywane do ponownego użycia.

Sumowanie tablic zrealizowano z wykorzystaniem metody kaskadowej redukcji, opisanej w pracy (Harris, 2008). Jest to metoda bardzo wydajnego sumowania danych, z wykorzystaniem przetwarzania równoległego z uwzględnieniem architektury procesorów GPU.

Na podstawie analizy zapisów wideo zawierających nagrania zabiegów bronchoskopowych można stwierdzić, że ruchy wykonywane endoskopem w trakcie „przeглядania” drzewa oskrzelowego przez lekarza, charakteryzują się niewielkimi prędkościami przesuwu i obrotu. Ograniczony zakres zmienności obrotów i translacji potwierdzają również wartości liczbowe, uzyskane z wykorzystaniem opracowanego algorytmu podczas analizy obrazów rzeczywistej bronchoskopii. Fakt ten został wykorzystany do ograniczenia liczby wykonywanych korelacji — współczynniki korelacji wyliczane są jedynie dla wąskiego zakresu przesunięć (parametr  $u$  w równaniu (4.29)). Dla macierzy przesunięć arbitralnie przyjętym zakresem było  $u \in [-50, 50]$ , zaś w przypadku obrotu  $u \in [-20, 20]$ .

Dane do korelowania umieszczane są w pamięci tekstur, do której odwołania są buforowane w pamięci podręcznej (ang. *cache*). Dodatkowo możliwe jest utworzenie z przesłanych danych bufora cyrkulacyjnego — odwołania do pikseli o indeksach  $x$  większych niż rozmiary obrazu  $N_x$  są zamieniane bądź na wartość piksela „na krawędzi” obrazu  $x' = N_x$ , bądź na wartość równą  $x' = \text{mod}(x, N_x)$ . Adresowanie *modulo* zostało wykorzystane przy korelowaniu fragmentów obrazów przy estymacji obrotu. Dane te „zapętlają się” w wierszach, dzięki temu podczas wyliczania współczynnika korelacji (licznika równania (4.29) zawsze wymnażane są wszystkie elementy macierzy. Dzięki temu współczynnik normalizujący może być obliczony jednokrotnie dla wszystkich przesunięć  $u$ .

W przypadku estymacji przesunięcia, wyniki interpolacji danych (rozwiniecie ścian tchawicy na cylinder) są najpierw transponowane, potem dzielone na mniejsze fragmenty, a następnie korelowane. Ponieważ w tym przypadku dane w wierszach nie są „zapętlone” (kolejne piksele wiersza znajdowały się coraz głębiej tchawicy), niezbędne jest ograniczenie rozmiarów korelowanych danych w taki sposób, by korelowany piksele „nie wyszedł” poza rozmiar wycinka obrazu. W tym celu algorytm koreluje jedynie  $N_x - 2 \cdot u$  pikseli w każdym wierszu. Zgodnie z definicją unormowanej korelacji wzajemnej obrazów, normalizacja powinna być przeprowadzana dla każdego przesunięcia  $(u, v)$  (równanie 4.29). W celu uproszczenia obliczeń przyjęto stałą wartość współczynnika normalizacji, wyznaczanego jednokrotnie dla każdego wydzielonego do korelacji fragmentu. Uproszczenie to nie ma wpływu na działanie algorytmu, ponieważ nie jest szukana wartość maksymalna korelacji, ale jej położenie w macierzy współczynników korelacji. Dodatkowo można zauważyć, że rozpatrywane przesunięcia są relatywnie niewielkie w porównaniu z rozmiarem korelowanego wycinka.

Uwzględnienie architektury procesora GPU poprzez optymalizację dostępu do pamięci, wykorzystanie pamięci współdzielonej oraz kaskadowego sumowania wartości, zapewniła wysoką wydajność algorytmu.

Autor zaimplementował również metodę korelowania obrazów z wykorzystaniem szybkiej transformaty Fouriera liczonej przez procesor karty graficznej. Metoda ta bazuje na bibliotece

CUFFT (NVIDIA, 2012), która jest częścią biblioteki CUDA. Umożliwia ona obliczanie transformaty Fouriera algorytmem Cooley-Tukey i Bluesteina dla danych rzeczywistych i zespolonych, zorganizowanych w macierze jedno-, dwu- lub trójwymiarowe.

W pierwszym kroku algorytmu wyznaczania korelacji (rozdziale 4.3.3) z wybranych fragmentów obrazu usuwana jest wartości średnia. Następnie, dane kopiowane są do wcześniej przygotowanych tablic buforowych (z dopisanymi zerami) przy jednoczesnej konwersji na liczby zespolone. Dzięki temu, możliwe jest obliczanie transformaty „*in-place*”, czyli w tej samej macierzy zapisywane jest widmo obrazu. Na tak przygotowanych danych obliczana jest dwuwymiarowa transformata Fouriera (funkcja *cufftExecC2C*). W kolejnym kroku, program CUDA wylicza sprzężenie macierzy oraz wymnaża odpowiadające sobie współczynniki (równanie 4.27). Ostatnim krokiem jest obliczenie transformaty odwrotnej. W wyniku obliczeń otrzymywana jest macierz współczynników korelacji, w której odszukiwana jest wartość maksymalnego współczynnika korelacji. Wyznaczone indeksy są miarą obrotu lub przesunięcia endoskopu.

### 4.6.3 Analiza wydajności

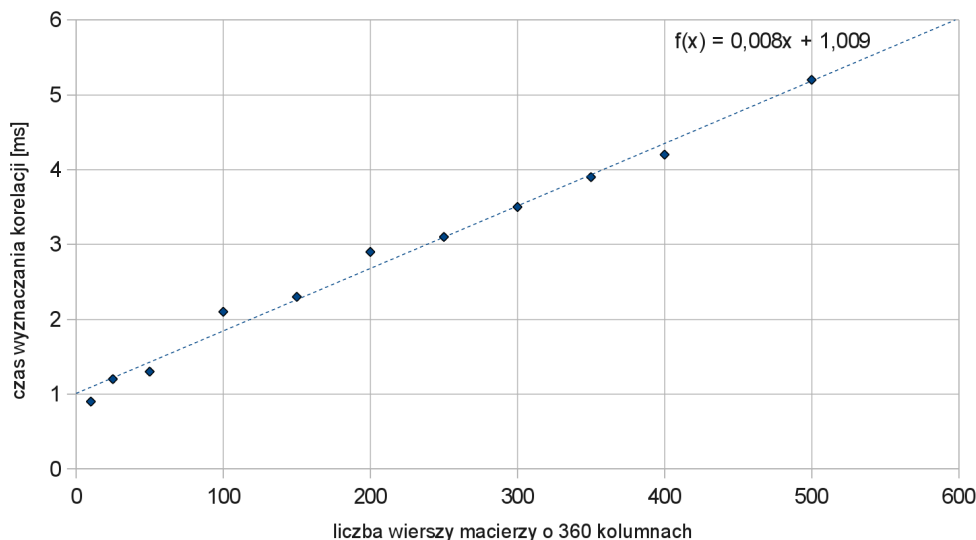
W celu łatwiejszego odniesienia się do wcześniej opublikowanych pomiarów wydajności algorytmu (Zieliński i inni, 2007) przy wykonywaniu nowych pomiarów, zachowano takie same nastawy algorytmu. Rozdzielczość obrazów wejściowych wynosiła 369 na 382 piksele, co daje 422874 bajtów dla każdego obrazu zapisanego w kolorach RGB oraz 140958 bajtów po konwersji do odcieni szarości. Obraz z kamery umieszczany był w tablicy wartości typu *uchar3* o rozmiarach 422874B. Macierz *float2* współrzędnych pikseli do korekcji zniekształceń miała rozmiar 563832B. Macierz do obliczenia przesunięcia kamery miała wymiary 500 na 360 pikseli, co daje tablicę wartości *float* o rozmiarze 720kB. Do wypełnienia tej tablicy (przez próbkowanie obrazu oryginalnego i interpolacji danych) potrzebna jest również tablica o takim samym rozmiarze zawierające dane typu *float2* (struktura łącząca dwie wartości *float*, współrzędne pikseli). Jej rozmiar jest równy 1,44MB. Macierz do estymacji obrotu kamery miała wymiary 100 na 360 pikseli, co daje tablicę *float* o rozmiarze 144kB oraz 288kB tablicę wartości *float2*. Liczba wyznaczanych wektorów dla ruchu postępowego tchawicy wynosiła 16.

Do oszacowania wydajności algorytmu wyznaczania korelacji z użyciem FFT przeprowadzono eksperyment, który polegał na pomiarze czasu korelacji dwóch macierzy o różnych rozmiarach. Przyjęto, że korelowane obrazy mają 360 kolumn natomiast liczba wierszy zmienia się od 10 do 500. Wymiary te odpowiadają rozmiarom macierzy w algorytmie estymacji ruchu endoskopu. Dla każdego badanego rozmiaru macierzy wykonano 20 powtórzeń. Wyniki pomiaru czasu zostały uśrednione i zaprezentowane na rysunku 4.15.

Można zauważyć, że uruchomienie i przygotowanie danych do obliczania trzech transformat Fouriera wymaga czasu około 1ms, niezależnie od rozmiarów korelowanych danych. Zastosowanie tej metody do wielokrotnego korelowania niewielkich wycinków danych, które ma miejsce zwłaszcza przy szacowaniu przesunięcia endoskopu, jest mało efektywne. Czas potrzebny do wyliczenia przesunięcia endoskopu z 16 fragmentów nie będzie mniejszy niż 16ms.

Wyniki pomiarów czasu wykonywania poszczególnych fragmentów algorytmu szacowania ruchu endoskopu zebrano w tabeli 4.3. Wszystkie eksperymenty przeprowadzono dla tego samego zbioru stu obrazów testowych pochodzących z bronchoskopu. Wyniki pomiarów zostały uśrednione. Zmierzony czas potrzebny do przesłania jednej ramki obrazu do pamięci wyniósł 0,03ms.

Głównym czynnikiem wyznaczającym maksymalną prędkość przesyłu danych między komputerem a kartą graficzną jest częstotliwość pracy magistrali danych oraz jej szerokość.



Rysunek 4.15: Zależność czasu korelowania macierzy 2D w funkcji liczby wierszy dla algorytmu używającego biblioteki CUFFT

Tablica 4.3: Czas wykonywania kluczowych fragmentów kodu dla opracowanego algorytmu szacowania ruchu endoskopu, uruchomionego na procesorze Core2Quad 2,4GHz i karcie graficznej z procesorem GeForce 260

Operacja	Czas obliczeń [ms]			
	Matlab	C++	CUDA	CUFFT
Korekcja nieliniowości obrazu	52	18	0,06	
Wykrycie rozbłysku	10	13	0,02	
Przygotowanie danych (interpolacja)	74	23	0,04	
Estymacja ruchu endoskopu (korelacja)	692	242	9,12	21

Drugim czynnikiem wpływającym na wydajność procesora jest liczba rdzeni CUDA. Użyta do testów karta graficzna posiadała procesor *GeForce GTX 260* z 216 rdzeniami CUDA, 448b magistralę oraz przepustowość pamięci (ang. *memory bandwidth*) na poziomie 111,9GB/s. Teoretyczna, maksymalna wydajność tego procesora sięga 1TFlop, czyli miliard operacji zmiennoprzecinkowych na sekundę.

Zgodnie z przypuszczeniami, w wyniku równoległego przetwarzania niewielkiej porcji danych, czasy wykonania poszczególnych fragmentów kodu algorytmu szacowania przesunięcia końcówki endoskopu uległy znacznemu skróceniu. Uzyskane przyśpieszenie działania algorytmu umożliwia szacowanie przesunięcia i obrotu kamery endoskopu w czasie krótszym niż 20ms, czyli przed nadejściem kolejnej ramki obrazu. Czas ten może być skrócony poprzez ograniczenie rozmiarów korelowanych macierzy. Jednocześnie użycie nowocześniejszej karty graficznej, o większej liczbie procesorów działających wydajniej, gwarantuje skrócenie czasu obliczeń.

Użycie algorytmu korelowania danych z wykorzystaniem szybkiej transformaty Fouriera zaimplementowanej w bibliotece CUFFT dało na testowej platformie sprzętowej około dziesięciokrotne przyśpieszenie. Jednak ze względu na duże koszty przygotowania danych i oblicza-

nia transformaty metoda ta jest zbyt wolna do zastosowania w przetwarzaniu danych w czasie rzeczywistym. Zastosowanie procesora GPU o większej liczbie rdzeni skróci czas obliczeń.

Testując wydajność opracowanych algorytmów implementowanych z wykorzystaniem mocy obliczeniowej procesora graficznego GPU, można było zauważyć występujące sporadycznie „zacięcia” algorytmu, objawiające się znacznie dłuższymi czasami obliczeń. Spowolnienia te spowodowane były dzieleniem mocy obliczeniowej GPU z innymi programami, np. interfejsem użytkownika w systemie Linux Ubuntu. Również jednoczesne wykorzystanie procesora karty graficznej do generowania obrazów wirtualnej bronchoskopii i estymacji ruchu endoskopu wymaga częstego przełączania zadań. W celu rozwiązania problemu współdzielenia zasobów karty graficznej, zasadne byłoby wykorzystanie kilku kart graficznych, które wykonywałyby oddzielne zadania.

## PODSUMOWANIE

### 5.1 Uzyskane wyniki końcowe

Celem pracy było opracowanie parametrycznego modelu drzewa oskrzelowego, umożliwiającego generowanie obrazów wirtualnej bronchoskopii, a następnie opracowanie i przetestowanie, z wykorzystaniem opracowanego modelu, algorytmów estymacji przesunięcia końcówki endoskopu. Tak postawione cele zostały spełnione, a uzyskane wyniki opublikowano w licznych pracach na forum krajowym, jak i zagranicznym. Autor rozprawy spotkał się z pozytywnymi komentarzami ze strony środowiska lekarskiego, zwłaszcza dotyczącymi dużego potencjału edukacyjnego opracowanego modelu zabiegu bronchoskopii.

**Opracowanie modelu zabiegu bronchoskopii.** Autor rozprawy opracował metodologię tworzenia parametrycznego modelu drzewa oskrzelowego, który obejmuje: tworzenie kształtu wirtualnych dróg oddechowych, dodawanie szczegółów anatomicznych oraz modelowanie wyglądu utworzonej trójwymiarowej powierzchni. Model drzewa oskrzelowego uzupełniony został o modele kamery endoskopowej i oświetlenia. Uwzględnia on występujące w rzeczywistym instrumencie medycznym zjawiska, takie jak: nieliniowe zniekształcenia obrazu wprowadzane przez szerokokątną optykę endoskopu, rozmycie obrazu wynikające ze skończonej głębi ostrości oraz gwałtownych ruchów endoskopu. Opracowany model wirtualnego oświetlenia, podążającego za wirtualną kamerą, uwzględnia jego kierunkowe działanie, zmianę intensywności w stożku światła oraz zjawisko zmniejszenia natężenia światła wraz z odległością. Połączenie modelu drzewa oskrzelowego z modelami kamery endoskopu i oświetlenia umożliwia generowanie obrazów wirtualnej bronchoskopii, które wykazują znaczne podobieństwo do obrazów rejestrowanych w trakcie rzeczywistych zabiegów bronchofiberoskopii.

Zdaniem autora, bardzo dużą zaletą opracowanego „fantomu” zabiegu bronchoskopii jest możliwość kontrolowania dużej liczby parametrów, które wpływają na ostateczny obraz wirtualnej bronchoskopii. Możliwe jest generowanie kształtu drzewa oskrzelowego o dowolnej topologii i złożoności, która kontrolowana jest przez niewielki zbiór punktów wyznaczających rozwidlenia. Tak przygotowany uproszczony szkielet dróg oddechowych jest następnie uszczegóławiany poprzez dodanie anatomicznie poprawnych łuków wygładzających rozwidlenia. Do szkieletu mogą być również dodawane dodatkowe szczegóły anatomiczne, takie jak przewęże-

nia czy spłaszczenia ścian. Na podstawie szczegółowego szkieletu generowane jest pole skalarne, które może być następnie dyskretyzowane z zadaną rozdzielczością przestrzenną. Dzięki temu, uzyskany model drzewa oskrzelowego może być traktowany tak, jak trójwymiarowa macierz danych uzyskanych np. z tomografii komputerowej. Rekonstrukcja powierzchni modelu drzewa oskrzelowego odbywa się z wykorzystaniem konturowania, tak samo jak ma to miejsce w przypadku odtwarzania powierzchni rzeczywistych struktur anatomicznych. Opracowany przez autora model drzewa oskrzelowego może więc być również używany do testowania i weryfikacji algorytmów przetwarzania danych medycznych, bazujących na danych wolumetrycznych.

Uzyskana przez konturowanie trójwymiarowej macierzy powierzchnia drzewa oskrzelowego ma postać gęstej siatki połączonych trójkątów, które mogą być w wydajny sposób wizualizowane z wykorzystaniem sprzętowych możliwości kart graficznych. Autor rozprawy zaimplementował i przetestował różne algorytmy obliczania koloru wirtualnych ścian drzewa oskrzelowego. Algorytm cieniowania powierzchni wpływa w sposób decydujący na to, w jaki sposób powierzchnia jest postrzegana. Metody najprostsze, takiej jak cieniowanie Phong, generują obrazy bardzo „sztuczne”, odbierane jako „plastikowe”. Jest to ich podstawowa wada. Jednak niewątpliwą zaletą najprostszych metod cieniowania są niskie koszty obliczeniowe. Dzięki temu możliwe jest generowanie nawet do tysiąca obrazów na sekundę. Taka wydajność jest niezbędna do generowania danych wejściowych dla wydajnych algorytmów dopasowywania obrazów (np. z wykorzystaniem informacji wzajemnej). Do takich zastosowań wystarczające są proste modele oświetlenia (np. cieniowanie Phong) uzupełnione o szczegóły na powierzchni, np. poprzez proceduralne mapowanie tekstur. Ze względu na wymaganą, zazwyczaj w takich zastosowaniach, niewielką rozdzielczość obrazów, proceduralny (np. cylindryczny) sposób mapowania tekstur jest akceptowalny, pomimo wprowadzania artefaktów.

Opracowany model drzewa oskrzelowego umożliwia również budowę „symulatorów” rzeczywistych zabiegów endoskopowych. Mogą one być użyte zarówno do zadań szkoleniowych, jak również do testowania i weryfikacji algorytmów. W takim przypadku, niezbędne jest generowanie obrazów wirtualnej bronchoskopii wyższej jakości z uwzględnieniem rzeczywistych zjawisk występujących podczas zabiegu. Dzięki możliwości wydajnego symulowania zniekształceń oraz rozmycia obrazu, możliwe jest interaktywne generowanie obrazów wirtualnej bronchoskopii na wydajnym komputerze klasy PC. Uzyskanie podobieństwa obrazów wirtualnych do rzeczywistych wymaga zastosowania bardziej zaawansowanych modeli oświetlenia oraz dokładniejszej metody teksturowania. Autor przetestował w tym celu możliwości tzw. mapowania UV. Przy użyciu programu Blender, wygenerował współrzędne mapowania dla konkretnego modelu drzewa oskrzelowego poprzez automatyczny podział i „rozwiniecie” powierzchni drzewa oskrzelowego na płaszczyznę. Na tej podstawie, używając programu graficznego Gimp, narysował teksturę drzewa oskrzelowego, na którą składały się wielokrotnie powielone fragmenty obrazów rzeczywistych tkanek. Siatka z przypisanymi współrzędnymi mapowania, wraz z odpowiadającą jej teksturą oraz zaawansowanym modelem cieniowania, została użyta jako wirtualny fantom zabiegu bronchoskopii do testowania algorytmów estymacji ruchu.

**Estymacja przesunięcia końcówki bronchofiberoskopu.** Druga część rozprawy dotyczy algorytmów estymacji ruchu końcówki bronchofiberoskopu. Autor rozprawy przeniósł implementację prototypowego systemu wspomaganego zabiegów bronchofiberoskopii, zaimplementowanego w środowisku Matlab, do kodu języka C++. Tak otrzymane oprogramowanie zostało użyte do przetestowania wydajności i skuteczności estymowania pozycji endoskopu. Autor

skupił się na kluczowym, z punktu widzenia wydajności całego systemu, algorytmie szybkiego szacowania przesunięcia endoskopu.

Zrealizowany w ramach prac zespołu pracowników Katedry Metrologii i Elektroniki, którego członkiem był również autor rozprawy, model systemu wspomagania zabiegu bronchoskopii swoje działanie opiera o dwa źródła danych: obrazy z endoskopu rejestrowane w trakcie zabiegu bronchoskopii oraz trójwymiarowe dane z tomografii komputerowej, na podstawie których rekonstruowana jest trójwymiarowa powierzchnia drzewa oskrzelowego pacjenta. Na podstawie tych danych określana jest aktualna pozycja endoskopu w drzewie oskrzelowym pacjenta. Informacja o pozycji endoskopu, względem np. węzłów chłonnych, jest istotna dla lekarza pulmonologa, którego zadaniem jest wykonanie biopsji powiększonych węzłów chłonnych. Obraz drzewa oskrzelowego uzyskiwany z endoskopu służy w pierwszym etapie do szacowania ruchu endoskopu. Drugim etapem działania algorytmu jest „dokręcanie” obrazu z wirtualnej kamery do obrazu rzeczywistego — zakłada się, że jeżeli obraz wirtualny jest podobny do obrazu rzeczywistego, to pozycje kamery wirtualnej i rzeczywistej się pokrywają. Znając współrzędne kamery wirtualnej, poruszającej się w zrekonstruowanym drzewie oskrzelowym pacjenta, można określić pozycję końcówki endoskopu.

Algorytm szacowania przesunięcia zakłada uproszczony model otoczenia endoskopu — tchawica traktowana jest jako walec, wewnątrz którego współosiowo porusza się końcówka endoskopu przekazująca szerokokątny obraz otoczenia. Model ruchu kamery został ograniczony do przesunięcia wzdłuż osi cylindra i obrotu wokół tej osi. Jak wynika z przeprowadzonych testów, tak prosty model zachowania endoskopu jest wystarczający do szacowania ruchu końcówki na podstawie kolejnych obrazów z kamery endoskopu.

Bazując na pracach zespołu, autor przebudował i zaimplementował algorytm szacowania przesunięcia endoskopu z wykorzystaniem technologii CUDA firmy NVIDIA. Umożliwia ona wykorzystanie w charakterze akceleratorów bardzo wydajnych procesorów graficznych GPU, wyposażonych w setki rdzeni. Autor przeniósł kosztowne obliczeniowo fragmenty algorytmu do procesora GPU karty graficznej. Zaimplementowany algorytm w pierwszym kroku przesyła obraz do pamięci karty graficznej, gdzie dokonywana jest pierwsza kosztowna obliczeniowo operacja, jaką jest korekcja nieliniowych zniekształceń szerokokątnego obrazu endoskopu. Operacja ta wymaga interpolacji danych. Zaimplementowana przez autora metoda wykorzystuje sprzętowe możliwości subpikselowego adresowania tekstur. W kolejnym kroku, na obrazie odszukiwana są współrzędne rozbłysku światła. Również ta operacja została zrealizowana z wykorzystaniem równoległego przetwarzania danych: obraz wymnżany jest z przygotowanym oknem Gaussa, zaś wartość maksymalna wyszukiwana jest w sposób kaskadowy. Na podstawie współrzędnych rozbłysku oraz stabelaryzowanych współrzędnych punktów do próbkowania, przygotowywane są macierze do estymacji przesunięcia i obrotu. Przesunięcie estymowane jest poprzez korelowanie obrazu fragmentów ścian drzewa oskrzelowego, zaś obrót obliczany jest na podstawie przesunięcia otoczenia rozbłysku rozwiniętego do układu biegunowego. Opracowany algorytm korelowania obrazów jest wersją algorytmu NCC (ang. *normalized cross correlation*), który zmodyfikowany został w taki sposób, by wykrywał jednowymiarowe przesunięcia dwuwymiarowych fragmentów obrazu. Algorytm uwzględnia również fakt, że dane wejściowe są obrazem w biegunowym układzie współrzędnych i tworzą w pamięci bufor cyrkulacyjny. Uwzględnienie wielordzeniowej architektury procesora GPU, zrównoleglenie obliczeń, optymalizacja dostępu do pamięci oraz wykorzystanie pamięci współdzielonej zapewniło wysoką wydajność algorytmu.

Uzyskany współczynnik przyspieszenia algorytmu w stosunku do wersji w C++ wyniósł około 26 razy, co umożliwia estymowanie przesunięcia endoskopu między kolejnymi klatkami



filmu na użytych do testów komputerze (procesor Intel Core 2 Quad Q6600 2,4GHz, karta graficzna z procesorem GeForce GTX 260). W przypadku implementacji w języku C++, czas przetwarzania danych wynosił około 0,3s i był około 2,8 razy krótszy niż w środowisku Matlab 2010a. Implementacja interpolacji oraz korelacji z wykorzystaniem procesora GPU karty graficznej, skróciło czas przetwarzania danych do 0,009s.

## 5.2 Oryginalne osiągnięcia

Za oryginalne elementy niniejszej pracy można uznać:

- opracowanie rozbudowanego modelu zabiegu bronchofiberoskopii, który obejmuje: metodologię konstruowania parametrycznego modelu drzewa oskrzelowego wraz z uwzględnieniem wyglądu jego powierzchni oraz parametryczny model endoskopu wraz z oświetleniem,
- implementację różnych modeli oświetlenia, które zostały następnie użyte do wygenerowania obrazów wirtualnej bronchoskopii, otrzymane obrazy oceniono pod kątem ich przydatności do symulacji wyglądu powierzchni drzewa oskrzelowego,
- zaimplementowanie w języku C++ prototypu systemu do wspomaganie zabiegu bronchoskopii zaprojektowanego w środowisku Matlab, który umożliwił oszacowanie kosztów obliczeniowych oraz umożliwił weryfikację poprawności działania
- opracowanie i zaimplementowanie w technologii CUDA kluczowych fragmentów algorytmu do szacowania przesunięcia końcówki endoskopu: szybkiej metody przygotowania i interpolacji obrazów oraz wyznaczania ich korelacji z uwzględnieniem architektury i możliwości akceleracji procesorów graficznych GPU,
- przetestowanie działania algorytmów estymacji ruchu endoskopu z wykorzystaniem obrazów rzeczywistych i wirtualnych, generowanych z użyciem opracowanego wirtualnego modelu zabiegu bronchofiberoskopii.



## ZAGADNIENIA INFORMATYCZNE

W rozdziale tym przybliżone zostaną najważniejsze narzędzia informatyczne, które zostały wykorzystane w implementacji wirtualnego fantomu oraz do algorytmów estymacji ruchu końcówki endoskopu. Omówiona zostanie biblioteka Visualization Toolkit (VTK), która została użyta do implementacji wirtualnego fantomu drzewa oskrzelowego. W dalszej części rozdziału omówiono możliwości akceleracji obliczeń przez równoległe przetwarzanie danych z wykorzystaniem karty graficznej oraz biblioteki CUDA™.

### A.1 Biblioteka VTK

Biblioteka *Visualization ToolKit*, w skrócie VTK, jest bardzo rozbudowanym narzędziem do przetwarzania oraz wizualizacji danych (Schroeder i inni, 2004; Kitware Inc., 2004; Socha, 2011b). Jest to wieloplatformowa biblioteka rozpowszechniana w postaci otwartego kodu źródłowego na zasadach wolnego oprogramowania (ang. *open-source*). Została ona zaimplementowana w języku C++ z bardzo intensywnym wykorzystaniem możliwości programowania zorientowanego obiektowo. Biblioteka zawiera również interfejsy do innych języków, takich jak Tcl/Tk, Java oraz Python i jest niezależna od graficznego interfejsu użytkownika (GUI). Zaimplementowano w niej wsparcie do popularnych bibliotek obsługujących graficzny interfejs użytkownika, takich jak: MFC, Qt czy Tk. Dzięki temu, możliwe jest umieszczenie kodu VTK w praktycznie dowolnym programie. Można jej użyć zarówno w programie typu konsola, jak i w programach okienkowych. Biblioteka dostępna jest w postaci skompilowanych bibliotek oraz „czystego” kodu źródłowego. Dostęp do kodu źródłowego biblioteki umożliwia skompilowanie jej w większości systemów operacyjnych (Windows, Linux, MacOS) przy użyciu dostępnych w danym środowisku kompilatorów. W trakcie konfiguracji możliwa jest zmiana wielu parametrów biblioteki (np. wybór między statyczną a dynamiczną wersją skompilowanej biblioteki czy też dołączenie dodatkowych narzędzi i bibliotek) (Martin i Hoffman, 2003).

Do wyświetlania grafiki biblioteka VTK używa interfejsu *OpenGL*, który umożliwia użycie bardzo wydajnych akceleratorów, wbudowanych w karty graficzne. Dzięki temu zapewnia bardzo dobrą wydajność oraz przenośność kodu. Implementacje OpenGL dostępne są dla większości platform sprzętowych (od superkomputerów, przez komputery klasy PC, po zaawansowane telefony komórkowe) oraz systemów operacyjnych. Biblioteka VTK powstała w firmie

Kitware i jest obecnie rozwijana przez szerokie grono ochotników oraz specjalistów z dziedziny grafiki komputerowej, algorytmów przetwarzania danych, wizualizacji, przetwarzania równoległego i wielu innych.

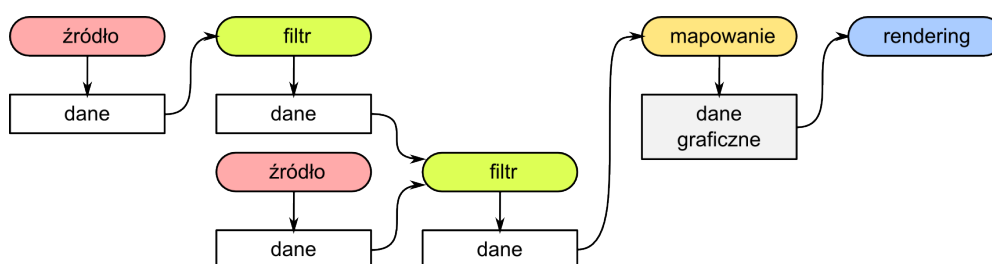
### A.1.1 Architektura

VTK jest biblioteką zorientowaną obiektowo, wykorzystującą paradygmat programowania obiektowego, w którym to program składa się z abstrakcyjnych *obiektów* łączących w sobie *dane* (atrybuty) z *metodami* (funkcjami), odpowiedzialnymi za przetwarzanie danych oraz komunikację z innymi obiektami. Takie podejście do programowania jest zgodne z intuicyjnym postrzeganiem rzeczywistych obiektów, które postrzegamy jako całość. Rzeczywiste obiekty posiadają atrybuty (cechy), które je definiują oraz funkcje, działania, które można wykonać przy pomocy lub *na* obiekcie.

Biblioteka VTK składa się z kilku tysięcy klas języka C++, intensywnie wykorzystujących możliwości tego języka. Niskopoziomowe części biblioteki, takie jak: struktury danych, algorytmy oraz funkcje wymagające dużej wydajności, zaimplementowane są w „czystym C++”. Powszechnie wykorzystywane są również *wzorce projektowe* (ang. *design pattern*), takie jak *fabryki obiektów* (ang. *object factories*), *obserwatory* (ang. *observer*), *iteratory* (ang. *iterator*) oraz funkcje wirtualne (ang. *virtual functions*), zapewniające polimorfizm obiektów. W wyniku tego biblioteka jest wydajna, łatwa w rozbudowie oraz niezależna od sprzętu.

### A.1.2 Strumień wizualizacji

Wizualizacja danych zaimplementowana w VTK wykorzystuje podejście strumieniowe, w którym dane przepływają między połączonymi obiektami. Przepływ danych (ang. *data flow*) zaczyna się od wejścia, jakim są obiekty *źródła* (ang. *source*) wczytujące lub generujące dane, poprzez obiekty *filtry* (ang. *filter*) odpowiedzialne za przetwarzanie danych, do obiektów *mapujących* (ang. *mapper*) wyniki przetwarzania do postaci podstawowych obiektów geometrycznych (np. punktów, linii) oraz ich parametrów (np. położenia, koloru czy rozmiaru). Na samym końcu strumienia znajdują się obiekty odpowiedzialne za generowanie obrazu (ang. *rendering*), czyli transformowania danych geometrycznych do postaci graficznej. Rysunek A.1 przedstawia przykładowy strumień wizualizacji (ang. *visualization pipeline*), zawierający: dwa źródła, filtry o jednym oraz dwóch wejściach, obiekt mapujący oraz renderujący obraz.



Rysunek A.1: Wzajemnie połączone obiekty tworzą strumień wizualizacji danych (Socha, 2011b)

Obiekty tworzące strumień wizualizacji danych, ze względu na odgrywaną w strumieniu rolę, można podzielić na dwie główne kategorie:

1. obiekty przechowujące abstrakcyjne dane z uwzględnieniem wzajemnych powiązań (topologii) oraz współrzędnych geometrycznych, będące potomkami obiektu `vtkDataObject`,
2. obiekty przetwarzające jedne dane w inne wywodzące się z klasy `vtkAlgorithm`, do których można zaliczyć:
  - źródła strumienia danych `vtkSource`, które obejmują obiekty wczytujące dane z plików i innych kanałów przesyłu informacji oraz obiekty generujące dane, takie jak np. generator parametrycznej sfery, którego wynikiem działania są współrzędne wierzchołków sfery o zadanej średnicy i liczbie wierzchołków,
  - filtry `vtkProcessObject` przetwarzające dane,
  - obiekty tworzące dane graficzne `vtkMapper`, zarówno dwu-, jak i trójwymiarowe, które stanowią pomost między strumieniem wizualizacji a częścią biblioteki odpowiedzialną za graficzną prezentację wyników ich działania,
  - inne wyjścia strumienia danych `vtkWriter`, głównie obiektów zapisujących dane do plików, które można potraktować jako specyficzne obiekty mapujące dane.

Obiekty VTK łączy się ze sobą zazwyczaj przy pomocy metod `SetInput()` oraz `GetOutput()`, np. wywołanie `Y.SetInput(X.GetOutput())` spowoduje połączenie wyjścia obiektu `X` do wejścia obiektu `Y`. Od wersji 5.0 biblioteki VTK, programiści przebudowali mechanizmy tworzenia, zarządzania i wykonywania strumienia wizualizacji. Nowy mechanizm jest kompatybilny wstecznie, działa jednak sprawniej i zapewnia jeszcze większą elastyczność. Podstawą nowego podejścia jest łączenie filtrów przy pomocy metod: `B->SetInputConnection(A->GetOutputPort())`. Różnica jest taka, że metoda `OutputPort()` (pochodząca od klasy `vtkAlgorithmOutput`) umożliwi realizację połączenia typu: „ $N$ -ty port  $X$ -owego algorytmu” obiektu `A`. Zazwyczaj  $N = 0$ , jednak można w jednym obiekcie w prosty sposób udostępnić wiele *portów*, przez które można dostarczyć różne dane z różnych części algorytmu. Zaś metoda `Output()` zwraca tylko jeden kontener (ang. *container*) na dane (Socha, 2011b).

Dane, które „płyną” w strumieniu wizualizacji mogą być różnego typu, dlatego konieczne jest sprawdzanie czy łączone ze sobą wejścia oraz wyjścia obiektów są ze sobą kompatybilne. Zgłaszanie błędów odbywa się podczas kompilacji programu w języku C++ lub podczas wykonywania programu w przypadku Tcl. VTK udostępnia również zaawansowane mechanizmy umożliwiające zarządzanie przechowywaniem danych w pamięci komputera. Jest to szczególnie ważne podczas wizualizacji dużych zbiorów danych, których rozmiary często przekraczają wielkość dostępnej pamięci operacyjnej. Możliwe jest włączenie lub wyłączenie wykorzystania pamięci podręcznej (ang. *cache*) oraz ograniczenie liczby kopii danych poprzez mechanizm zliczania referencji (ang. *reference counting*). Umożliwia on wykorzystanie jednej kopii danych jednocześnie przez kilka obiektów. W bibliotece zaimplementowano również dynamiczne zarządzanie pamięcią, które umożliwia natychmiastowe zwalnianie nieużywanych zasobów. Dostępne jest również statyczne zarządzanie pamięcią, które ogranicza potrzebę wielokrotnego dokonywania obliczeń pośrednich, jednak sprawdza się ono tylko przy przechowywaniu relatywnie małych porcji danych. Ostatnim mechanizmem wpływającym na wydajność i ograniczającym zapotrzebowanie na pamięć jest mechanizm strumieniowania danych, umożliwiający dzielenie danych na mniejsze części, podlegające obliczeniom. Mechanizm ten jest bardzo wydajny i umożliwia wykonywanie np. równoległych obliczeń na wielu niezależnych komputerach. Do komunikacji między nimi może być wykorzystywany interfejs OpenMP.

Biblioteka VTK posiada dwa tryby uruchamiania strumienia przetwarzania i wizualizacji danych:

- *zależny od zapotrzebowania na dane* (ang. *demand-driven*) — realizowany „od końca” strumienia przy pomocy metody `Update`, tylko gdy wyjście danego obiektu jest potrzebne. Dzięki temu, uruchamiana jest wyłącznie niezbędna część strumienia. Umożliwia to wykonywanie wielu zmian w strumieniu, bez konieczności wielokrotnego przeliczania danych. Ogranicza to zbędne obliczenia, co jest bardzo istotne w programach dostarczających interaktywnej wizualizacji danych,
- *sterowany zdarzeniami* (ang. *event-driven*) — realizowany „do końca” strumienia metodą `ExecuteData()`, której główną zaletą jest zapewnianie aktualności danych w strumieniu.

VTK przetwarza dane tylko wtedy, gdy są one potrzebne. Oznacza to, że samo połączenie w strumień obiektów nie „uruchomi” strumienia. Zarządzanie uruchamianiem poszczególnych obiektów strumienia odbywa się na podstawie przypisywanego indywidualnie dla poszczególnych obiektów tzw. *czasu modyfikacji obiektu*. Dzięki temu, możliwe jest określenie, czy konkretny obiekt ma dane nadal aktualne w porównaniu z innym obiektem. Wszystkie obiekty tworzące strumień wizualizacji posiadają metodę `Update()`, której zadaniem jest „uruchamianie” działania obiektów. Zazwyczaj nie ma potrzeby ręcznego uruchamiania obiektów, poprzez wywoływanie ich implementacji `Update()`, ponieważ odpowiadają za to automatyczne mechanizmy strumienia wizualizacji. Jak już wcześniej wspomniano, VTK przetwarza dane tylko wtedy, gdy te dane są potrzebne, a poprzednie wyniki obliczeń są nieaktualne. „Zapotrzebowanie” automatycznie zgłaszają obiekty, znajdujące się na końcu strumienia: *mapujące lub zapisujące dane*. Przepływ danych odbywa się w kierunku: *źródła* → *filtry* → *obiekty mapujące* → *obiekty generujące obraz*, zaś uruchamianie przetwarzania danych odbywa się w kierunku przeciwnym.

### A.1.3 Model graficzny

Strumień wizualizacji (ang. *visualization pipeline*) stanowi podstawę biblioteki VTK i jest głównym miejscem, gdzie dokonywane jest przetwarzanie danych. Drugim bardzo ważnym podsystemem jest część odpowiedzialna za *tworzenie i prezentację danych graficznych* w postaci cyfrowego obrazu. Ta część biblioteki, nazywana *modelem graficznym* (ang. *graphics model*), stanowi abstrakcyjną warstwę pomiędzy strumieniem wizualizacji a niskopoziomowymi interfejsami lub językami graficznymi.

Podstawowym interfejsem graficznym dla biblioteki VTK jest obecnie *OpenGL*. Jest to niezależna od systemu operacyjnego oraz sprzętu biblioteka, która do generowania cyfrowych obrazów wykorzystuje możliwości sprzętowej akceleracji generowania i przetwarzania grafiki na kartach graficznych. Oprócz biblioteki *OpenGL*, VTK umożliwia wykorzystanie biblioteki *Mesa*, będącej implementacją *OpenGL* z otwartym kodem. Kolejnym interfejsem VTK jest możliwość tworzenie obrazów przy pomocy „zewnętrznego” oprogramowania do renderingu, zgodnego z programem *RenderMan* lub *Povray*.

Wynikiem działania algorytmów mapujących są *dane graficzne* zbudowane z tzw. podstawowych obiektów graficznych (ang. *graphics primitives*). W zależności od implementacji obiektu mapującego, lista graficznych obiektów podstawowych może być różna. Ponieważ głównym interfejsem graficznym VTK jest *OpenGL*, podstawowe obiekty graficzne pokrywają się z obiektami biblioteki *OpenGL* i są to:

- punkt (ang. *point, vertex*) (`GL_POINTS`) posiada zestaw parametrów, takich jak: położenie (współrzędne), kolor oraz wektor normalny,
- linia (ang. *line*) (`GL_LINE`) łączy dwa punkty, tworzy krawędź,
- krzywa (ang. *polyline*) (`GL_LINE_STRIP`), będąca ciągiem połączonych linii,

- krzywa zamknięta (GL\_LINE\_LOOP) zbudowana jest z serii połączonych linii, w których ostatni punkt jest połączony z pierwszym,
- trójkąt (ang. *triangle*) (GL\_TRIANGLES) rozpięty na trzech punktach, posiada trzy krawędzie,
- pas trójkątów (ang. *triangle strip*) (GL\_TRIANGLE\_STRIP) jest zbiorem trójkątów, w których sąsiadujące trójkąty mają wspólne krawędzie,
- wachlarz trójkątów (ang. *triangle fan*) (GL\_TRIANGLE\_FAN) jest zbiorem trójkątów z jednym wspólnym wierzchołkiem,
- czworokąt (ang. *quad*) (GL\_QUADS) rozpięty na czterech punktach, rozmieszczonych na jednej płaszczyźnie,
- pas czworokątów (ang. *quad strip*) (GL\_QUAD\_STRIP) to zbiór czworokątów, którym sąsiadujące czworokąty posiadają wspólną krawędź,
- wielokąt (ang. *polygon*) (GL\_POLYGON) to zbiór krawędzi tworzących zamknięty obszar, zazwyczaj rozpięty na punktach leżących na jednej płaszczyźnie, trójkąt i czworokąt są przykładami wielokąta.

Podstawowe obiekty graficzne stanowią cegiełki, z których podczas renderingu budowany jest cyfrowy obraz. Współczesne karty graficzne, zgodne ze standardem OpenGL, zapewniają w pełni sprzętową obróbkę obiektów podstawowych. Obejmuje ona obliczenia dotyczące transformowania współrzędnych wierzchołków (skalowanie, przesuwanie czy rotację), jak również wszystkie etapy „rysowania” obiektów podstawowych. Wydajność przetwarzania różnych obiektów podstawowych dla poszczególnych typów kart graficznych może być znacznie różna. Dzieje się tak, ponieważ procesory graficzne są zwykle optymalizowane do bardzo wydajnego przetwarzania ograniczonej liczby obiektów podstawowych, najczęściej jest to tylko lista trójkątów.

#### A.1.4 Podstawowe obiekty modelu graficznego VTK

Przyjęty w VTK model graficzny wykorzystuje klasy, których nazewnictwo nawiązuje do przemysłu filmowego. Podstawowym pojęciem jest wirtualna *scena*, na której znajdują się *aktorzy*, będący wynikiem działania strumienia wizualizacji, *światła* oraz *kamery*. Widok z wirtualnej kamery prezentowany jest w oknach wizualizacji.

Do najważniejszych klas *modelu graficznego* należą:

- vtkActor, vtkActor2D, vtkVolume** — klasy używane do reprezentowania *danych graficznych* (czyli wyjścia z obiektów mapujących) na wirtualnej scenie.
- vtkLight** — obiekty reprezentujące wirtualne światło, zarówno równoległe (z nieskończoności), jak i punktowe (pozycyjne). Posiadają, takie parametry jak: jasność, kolor, typ, kąt stożka światła (dla punktowego), współczynniki tłumienia wraz z odległością oraz znikania. Dodatkowo mają macierze transformacji, określające miejsce w przestrzeni, gdzie się znajdują.
- vtkCamera** — wirtualna kamera, służąca do określenia punktu w przestrzeni, z którego widziana jest scena (ang. *view point*) oraz punktu, na który kamera patrzy (ang. *focal point*). Dostępne metody umożliwiają, między innymi: określenie i zmianę współrzędnych kamery, zmianę szerokości kąta patrzenia oraz typu projekcji (ortogonalna lub perspektywiczna).
- vtkProperty, vtkProperty2D** — określa właściwości optyczne powierzchni obiektów graficznych, takie jak: kolory (ogólny, otaczający, rozproszony, połysku, krawędzi), współczynniki modelu oświetlenia (np. odbicia światła ang. *specular power*, rozproszenia ang. *diffuse* oraz współczynnik natężenia światła otoczenia ang. *ambient*), przeźroczystość, re-

prezentację geometrii obiektu (punkty, krawędzie, powierzchnie) oraz metodę cieniowania (płaskie, Gourauda, Phonga).

**vtkMapper, vtkMapper2D** — są to klasy abstrakcyjne, z których wywodzą się klasy stanowiące pomost między wizualizowanymi danymi a reprezentującymi je w postaci graficznej wirtualnymi aktorami.

**vtkTransform** — obiekty te mogą być wykorzystywane do transformacji afinicznych w trzech wymiarach opisanych przez macierz przekształceń jednorodnych (4x4).

**vtkLookupTable, vtkColorTransferFunction** — obiekty używane do tworzenia tzw. tablic kolorów, służących do mapowania danych skalarnych (indeksów tablicy) do odpowiadających im wartości koloru. Kolor może być kodowany zarówno w postaci triady RGB, z uwzględnieniem kanału przezroczystości RGBA oraz w systemie HSV.

**vtkRenderer** — jest obiektem, kontrolującym proces tworzenia obrazu. Rendering to proces przetwarzania danych geometrycznych, wirtualnego światła i widoku z kamery na cyfrowy obraz. Wykonuje również transformacje między układami współrzędnych: globalnym opisującym przestrzeń sceny, lokalnymi poszczególnych aktorów oraz współrzędnych wyświetlania na ekranie.

**vtkRenderWindow** — abstrakcyjny obiekt, którego implementacje umożliwiają umieszczenie okna z wynikami wizualizacji w interfejsie użytkownika. Metody obiektu umożliwiają synchronizację, zmianę rozmiarów (rozdzielczości) okna wizualizacji, zarządzanie buforowaniem oraz tworzeniem obrazów stereowizyjnych.

**vtkRenderWindowInteractor** — stanowi niezależny od platformy mechanizm interakcji z oknem wizualizacji. Klasa ta, wspólnie z klasą **vtkInteractorStyle** umożliwia interaktywne sterowanie wizualizacją, np. ruchy wykonywane myszką mogą sterować położeniem kamery. W zależności od konkretnej implementacji stosowanej klasy **vtkInteractorStyle** źródłem interakcji może być użytkownik (poprzez myszkę, klawiaturę lub inny kontroler), jak również inne obiekty (np. licznik czasu (ang. *timer*) sterujący prędkością animacji).

Kombinacja powyższych obiektów, w połączeniu z wynikami działania strumienia wizualizacji, umożliwia uzyskanie interaktywnej wizualizacji danych wejściowych.

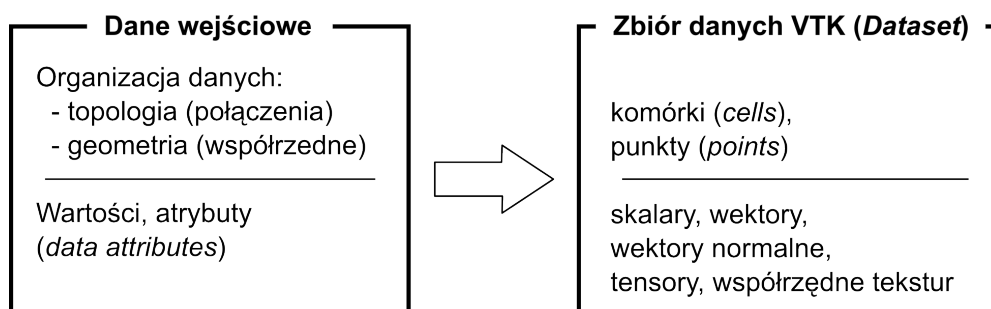
### A.1.5 Reprezentacja danych

Postrzegany przez nas rzeczywisty świat jest zazwyczaj ciągły, zaś dane, szczególnie w postaci cyfrowej, mają charakter dyskretny. Dane opisujące zjawiska czy też obiekty posiadają *atomy*, czyli charakteryzujące je wartości. Wizualizacja ciągłych zjawisk, opisanych dyskretnymi wartościami, wymaga więc zastosowania *interpolacji*. Do interpolacji wykorzystuje się *topologię*, czyli informację o tym, które dane i w jaki sposób są ze sobą połączone. Topologia danych opisuje połączenia niezależnie od przeprowadzanych przekształceń geometrycznych. Dzięki zdefiniowanej topologii możliwe jest wyznaczenie wartości pośrednich. Zmiana topologii połączeń prowadzi do zmian w sposobie interpolacji danych, a tym samym zmienia efekt wizualizacji (Socha, 2011b). Oprócz topologii, dane zawierają również *informacje geometryczne* o ich rozmieszczeniu w przestrzeni.

Zastosowany w VTK *model danych* (ang. *dataset*) przyjmuje, że dowolnym, abstrakcyjnym danym, można przypisać dwa rodzaje informacji: *wartości* oraz skojarzoną z nimi *strukturę danych*. Model ten przedstawiony jest na rysunku A.2. Struktura lub inaczej organizacja danych opisana jest przez *topologię* (połączenia) oraz *geometrię* (współrzędne). Do przechowywania informacji o strukturze danych VTK używa licznej rodziny obiektów tzw. *komórek* (ang. *cells*). Komórki stanowią pomost pomiędzy danymi dyskretnymi a ich interpolacją. Dla danych bez



topologii informacje przechowywane są w punktach. Z kolei *wartości* (ang. *data attributes*) mogą mieć różną reprezentację, mogą to być: wartości skalarne, wektorowe, tensorowe, tekstury, itp.



Rysunek A.2: Dane wejściowe i odpowiadający im model danych VTK (ang. *dataset*) (Socha, 2011b)

Przyjęty model danych, w którym wyróżnia się: wartości, topologię oraz geometrię, reprezentowane przez zbiór punktów (ang. *points*) i komórek (ang. *cells*), jest odzwierciedleniem dyskretnej natury danych. Punkty określają jedynie miejsce, w których znamy wartość ciągłych zjawisk, zaś komórki umożliwiają dokonanie interpolacji między punktami. Komórki definiowane są poprzez określenie *typu*, a następnie przypisanie uporządkowanej listy punktów, tzw. *listy połączeń* (ang. *connectivity list*).

Biblioteka VTK ma zaimplementowanych wiele typów komórek. Podstawowym kryterium podziału jest *wymiar topologiczny komórki*. Określa on najmniejszą liczbę potrzebnych do opisanego współrzędnych punktu w przestrzeni  $N$ -wymiarowej: punkty posiadają wymiar 0, krzywe mają wymiar 1 (tylko jedna liczba wystarczy, by określić, gdzie się znajduje dowolny punkt na krzywej), powierzchnie mają wymiar topologiczny równy 2, zaś bryły geometryczne wymiar topologiczny 3. Innym podziałem może być zastosowana funkcja interpolująca, która może być: liniowa lub nieliniowa.

Komórki definiowane są poprzez: *typ* oraz posortowaną *listę połączeń* (ang. *connectivity list*). Lista połączeń zawiera indeksy punktów, te zaś prowadzą do listy punktów zawierającej ich współrzędne geometryczne.

Na rysunkach A.3 oraz A.4 przedstawiono podstawowe typy komórek VTK, które używają liniowej interpolacji danych między wierzchołkami. Do obiektów o zerowym wymiarze topologicznym zalicza się:

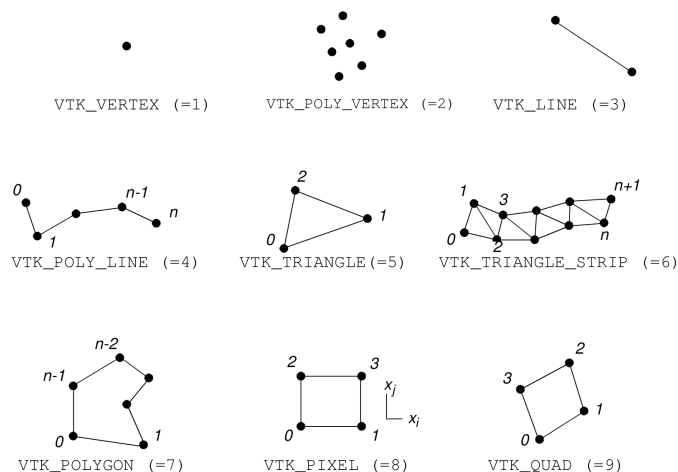
- wierzchołek VTK\_VERTEX jest najprostszym i podstawowym typem komórki, wyznacza jeden punkt w przestrzeni, niesie informację jedynie o położeniu geometrycznym bez definiowania połączeń,
- zbiór wierzchołków VTK\_POLY\_VERTEX definiowany jest poprzez listę punktów w przestrzeni,

Jednowymiarowymi topologicznie (1D) komórkami VTK są:

- linia VTK\_LINE — podstawowy typ jednowymiarowy, definiowany przez dwa punkty, przy czym kierunek wyznaczany jest od punktu pierwszego do drugiego,
- krzywa VTK\_POLY\_LINE — zbiór linii, definiowany przez uporządkowaną listę  $n+1$  punktów, gdzie  $n$  jest liczbą linii tworzących krzywą, pary punktów  $(i, i+1)$  wyznaczają poszczególne linie,

Komórki dwuwymiarowe (2D) to:

- trójkąt VTK\_TRIANGLE — podstawowy obiekt o dwuwymiarowej topologii, definiowany przez listę trzech punktów, kolejność punktów wyznacza kierunek wektora normalnego (zasada prawej dłoni — jeżeli palce wskazują kierunek wzrostu indeksów punktów, to odchylony kciuk wyznacza kierunek wektora normalnego),
- pas trójkątów VTK\_TRIANGLE\_STRIP — zbiór połączonych trójkątów, wyznaczany przez uporządkowaną listę  $n + 2$  punktów, gdzie  $n$  jest liczbą trójkątów; pojedynczy trójkąt definiują trzy kolejne punkty listy połączeń:  $(i, i + 1, i + 2)$ , gdzie  $0 \leq i \leq n$ , punkty nie muszą leżeć na jednej płaszczyźnie,
- czworokąt VTK\_QUAD — wyznaczany przez uporządkowaną listę czterech punktów, leżących na jednej płaszczyźnie i tworzących figurę wypukłą, o nieprzecinających się krawędziach; wektor normalny wyznaczany jest podobnie jak w przypadku trójkąta, poprzez kolejność punktów (przeciwną do kierunku ruchu wskazówek zegara) i zgodnie z regułą prawej dłoni,
- piksel VTK\_PIXEL — wyznaczany przez cztery punkty położone na jednej płaszczyźnie i tworzące prostokąt, którego równoległe krawędzie pokrywają się z wybranymi osiami układu współrzędnych, dodatkowo punkty sortowane są w taki sposób, by ich indeksy rosły wraz ze wzrostem współrzędnych układu odniesienia; komórka tego typu stanowi zoptymalizowaną obliczeniowo wersję czworokąta. Należy zwrócić uwagę na nazwę „*piksel*”, która zazwyczaj definiowana jest trochę inaczej, jako „*pojedynczy element obrazu*”. W przypadku komórki typu *piksel* zawiera ona cztery „elementy obrazu”.
- wielokąt VTK\_POLYGON — zamknięta figura, definiowana przez uporządkowaną listę więcej niż trzech punktów leżących na jednej płaszczyźnie; wielokąt nie musi być wypukły, nie mogą jednak występować w nim wewnętrzne zapętlenia oraz przecięcia krawędzi; wielokąt zawiera  $n$  krawędzi, gdzie  $n$  to liczba wierzchołków; wektor normalny wyznaczany jest w sposób bezpośredni, na podstawie odwrotnej do ruchu wskazówek zegara kolejności punktów i zasadzie prawej dłoni,



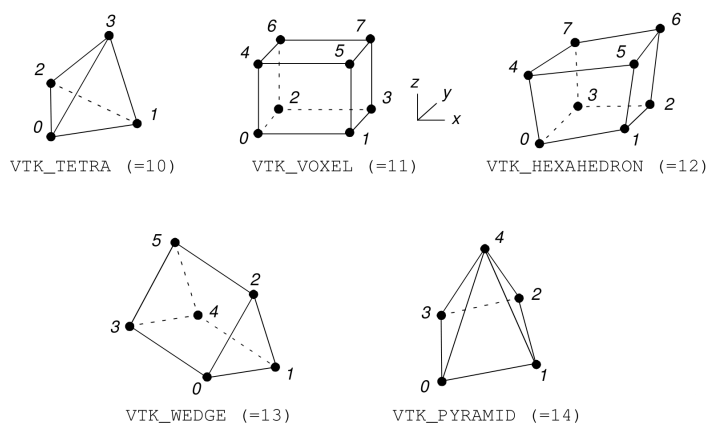
Rysunek A.3: Przykłady komórek o zerowych wymiarach topologicznych (1,2), 1D (3,4) oraz 2D (5-9), nazwy i oznaczenia na podstawie vtkCell (Schroeder i inni, 2004)

Na rysunku A.4 przedstawiono komórki o topologii trójwymiarowe (3D):

- czworościan VTK\_TETRA definiuje lista czterech punktów, które nie leżą na jednej płaszczyźnie; czworobok posiada sześć krawędzi oraz cztery wierzchołki,
- sześciocian VTK\_HEXAHEDRON definiuje lista ośmiu punktów, nie leżących na jednej płaszczyźnie; zbudowany jest z sześciu czworokątnych ścian, dwunastu krawędzi oraz

ośmiu wierzchołków, niedozwolone są przecięcia ścian i krawędzi, sześciścian musi być wypukły,

- woksel VTK\_VOXEL jest optymalizowaną obliczeniowo wersją sześciścianu, pokrywającą się z nim topologicznie, jednak z narzuconymi ograniczeniami geometrycznymi: wszystkie krawędzie oraz ściany muszą być prostopadłe lub równoległe do osi układu współrzędnych; dodatkowo, lista punktów musi być posortowana tak, by indeksy punktów wzrastały wraz ze wzrostem wartości współrzędnych układu odniesienia (rysunek A.4). Podobnie jak w przypadku typu piksel, ten typ komórki definiuje osiem „pojedynczych elementów” obrazu przestrzennego (ang. *volume data*), które są rozmieszczone „w rogach” komórki typu *woksel*,
- klin VTK\_WEDGE jest komórką definiowaną listą sześciu punktów połączonych dziewięcioma krawędziami, które tworzą dwie ściany trójkątne oraz trzy czworokątne, niedozwolone są przecięcia krawędzi i ścian, figura musi być wypukła,
- piramida VTK\_PYRAMID definiowana jest przez listę pięciu punktów, pierwsze cztery leżą na jednej płaszczyźnie i tworzą czworokątną ścianę, piąty zaś wyznacza z pozostałymi cztery ścianki trójkątne (rysunek A.4).



Rysunek A.4: Przykłady komórek o trójwymiarowej topologii (Schroeder i inni, 2004)

Komórki jednowymiarowe stosowane są zazwyczaj do oznaczania konturów i krawędzi. Piksele, ze względu na dużą wydajność wynikającą z prostej budowy, stanowią bazę do przechowywania cyfrowych obrazów (zwykle regularnych macierzy danych dwuwymiarowych). Natomiast woksele są wykorzystywane do reprezentacji danych medycznych (trójwymiarowych macierzy wyników, np. z tomografii komputerowej). Z kolei nieregularne komórki dwuwymiarowe (trójkąt, czworokąt) oraz trójwymiarowe (czworościan i sześciścian) są szczególnie chętnie stosowane w symulacjach komputerowych, zwłaszcza w tzw. *metodach elementów skończonych* (ang. *FEM, finite-element method*), mechanice płynów i innych.

### A.1.6 Typy atrybutów danych

Wartości lub też inaczej *atrybuty* (ang. *attribute data*) są informacjami powiązаныmi ze strukturą danych: z geometrią i topologią. Wartości mogą być przypisane zarówno do punktów, jak i do komórek oraz ich części (krawędzi i ścianek). Przykładem atrybutów mogą być: wartości temperatury (skalar) lub prędkości (wektor) przypisane do punktów, masa przypisana do komórki lub strumień ciepła przypisany do ścianki komórki (Socha, 2011b).

Biblioteka VTK dzieli atrybuty danych na następujące typy:

- skalary (ang. *scalars*) — dane posiadające tylko jedną wartość w każdym punkcie, jest to najczęściej spotykany typ danych. Przykładem może być: temperatura, ciśnienie, gęstość czy cena.
- wektory (ang. *vectors*) — dane posiadające: wartość, kierunek oraz zwrot. W przestrzeni dwuwymiarowej określone przez dwie składowe ( $u, v$ ), zaś w trójwymiarowej przez trzy ( $u, v, w$ ). Przykładem jest prędkość lub gradient temperatur.
- wektory normalne (ang. *normals*) — wektor jednostkowy o długości równej 1. Stosowane głównie do obliczeń kolorów powierzchni oświetlonych oraz określenia ich orientacji przestrzennej.
- współrzędne tekstur (ang. *texture coordinates*) — służą do mapowania punktu z kartezjańskiego układu współrzędnych do przestrzeni 1, 2 lub 3 wymiarowej tekstury. Tekstura jest regularną siatką (zazwyczaj obrazem), zawierającą informacje o kolorze, intensywności, często również o przezroczystości. Współrzędne tekstur umożliwiają wizualizację np. globusa, poprzez nałożenie dwuwymiarowej grafiki prezentującej mapę ziemi na trójwymiarową kulę.
- tensory (ang. *tensors*) — stanowią matematyczne uogólnienie wektora i macierzy. Wielkości tensorowe reprezentuje się w postaci macierzy kwadratowej, tensor rzędu  $k$  jest  $k$ -wymiarową macierzą kwadratową. Tensor rzędu zerowego to skalar, rzędu pierwszego to wektor zaś drugiego to macierz. Przykładami wielkości tensorowych są tensory naprężeń i odkształceń mechanicznych.

Obiektem VTK o szczególnym znaczeniu jest abstrakcyjna superklasa `vtkAbstractArray`, która stanowi protoplastę „abstrakcyjnego pojemnika” na dowolne dane, przechowywane w pamięci w postaci obszaru ciągłego. W obiektach wywodzących się z tej klasy przechowywane są wszelkiego rodzaju informacje zarówno *atrybuty* (wartości) danych, jak i dane dotyczące *topologii* (np. listy połączeń) i *geometrii* (np. współrzędne). Z tej abstrakcyjnej klasy wywodzą się dwie klasy służące do przechowywania danych tekstowych `vtkStringArray` i `vtkUnicodeStringArray` oraz klasa `vtkDataArray`.

`vtkDataArray` jest nadklasą dla *danych numerycznych* przechowywanych w postaci tablic danych i stanowi podstawowe narzędzie do przechowywania i zarządzania tymi danymi. Z klasy tej wywodzi się duża grupa obiektów VTK, które są wyspecjalizowane do obsługi danych w tzw. *formatach podstawowych* (`bit`, `char`, `int`, `uint`, `float` itp.). Klasy te często mają zaimplementowane metody zoptymalizowane dla konkretnych typów danych, np. wstawianie, dodawanie i kasowanie danych, szybkie adresowanie czy też kopiowania itp. Zarządzanie pamięcią odbywa się w sposób dynamiczny. Obiekty przechowują *wskaźnik do adresu tablicy* danych konkretnego typu (np. `int`, `float`), aktualny rozmiar zarezerwowanej i wykorzystanej pamięci oraz *wskaźnik do adresu tablicy znaków*, przechowującej tekstową *nazwę* danych.

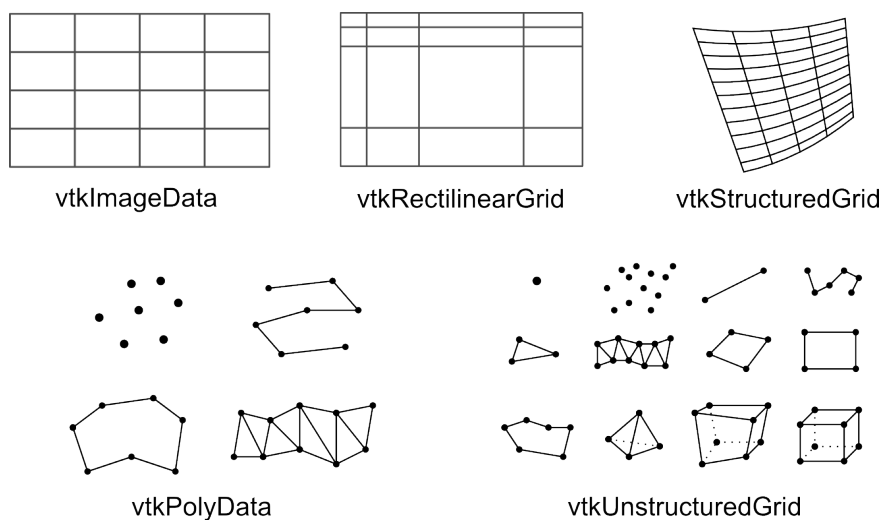
W pakiecie VTK zastosowano liniową organizację danych (wskaźnik do adresu tablicy danych). Nie jest ona wygodna, gdy niezbędne jest przechowywanie struktur bardziej skomplikowanych, takich jak np. cztery liczby całkowite opisujące kolor w modelu RGBA lub trzy liczby zmiennoprzecinkowe opisujące współrzędne punktu w trójwymiarowej przestrzeni. W celu ułatwienia dostępu do przechowywanych danych jednego typu, tworzących mniejsze grupy, zaimplementowano mechanizm oparty o tzw. *krotki* (ang. *tuple*)\*. Ciągła przestrzeń tablicy jest dzielona na mniejsze podciągi (krotki) o zadanej liczbie elementów (ang. *components*). Rozmiar krotki określany jest metodą `SetNumberOfComponents()`, zaś liczba krotek w tablicy danych metodą `SetNumberOfTuples()`. Dla wartości skalarnych krotka ma rozmiar 1, dla wektora trójwymiarowego 3, zaś dla tensora trzeciego rzędu 9. Rozmiar krotki jest stały dla wszystkich wartości w jednej tablicy.

---

\*VTK dopuszcza tylko jeden typ danych przechowywanych w krotce.

### A.1.7 Typy zbiorów danych

Przyjęty w VTK model danych łączy organizację danych z jej wartościami. Ze względu na różną organizację danych, wyróżnić można kilka podstawowych *typów zbiorów danych* (rysunek A.5):



Rysunek A.5: Podstawowe typy danych (ang. *types of dataset*) oraz obiekty biblioteki VTK umożliwiające ich obsługę (Schroeder i inni, 2004)

**Image Data** — dane, wyróżniające się równomiernym rozmieszczeniem punktów i komórek na regularnej siatce. Dane zorganizowane są w tabele: wiersze, kolumny oraz poziomy (warstwy), które są równoległe z układem współrzędnych. Dane tego typu mogą zawierać jeden wiersz danych (jednowymiarową linię elementów, np. paletę kolorów), piksele (jak np. typowy obraz dwuwymiarowy) oraz woksele (trójwymiarowe dane medyczne, np. z tomografii komputerowej). Ponieważ dane są zorganizowane regularnie zarówno topologicznie, jak i geometrycznie, współrzędne poszczególnych danych można wyznaczyć w sposób pośredni, co zmniejsza zapotrzebowanie na pamięć. Do wyznaczenia współrzędnych niezbędna jest informacja o *rozmiarach danych* (ang. *data dimension*), w przypadku obrazu dwuwymiarowego jest to jego rozdzielczość, *odległość między danymi* (ang. *data spacing*) oraz *punkt początkowy* (ang. *origin point*).

Rozmiary danych przechowywane są w postaci trójelementowego wektora  $(n_x, n_y, n_z)$ , który zawiera liczbę punktów w poszczególnych kierunkach (wymiarach). Punkt początkowy (ang. *origin point*) to wektor wyznaczający współrzędne pierwszego punktu danych, określone w trójwymiarowej przestrzeni sceny. Odległość między danymi wyznacza rozmiary pojedynczej komórki. Liczba punktów w danym kierunku pomnożona przez odległość między punktami (ang. *spacing*) wyznacza rozmiar geometryczny danych.

Obiekt VTK obsługujący te dane to `vtkImageData`. Podstawowym zastosowaniem tego typu danych jest przechowywanie regularnych macierzy (np. danych obrazowych), w tym danych medycznych, takich jak: obrazy z tomografii komputerowej czy rezonansu magnetycznego.

**Rectilinear Grid** — dane zorganizowane regularnie topologicznie, podobnie do danych typu *Image Data*, jednak z jedynie częściowo regularną geometrią. Odległości między poszczególnymi wierszami, kolumnami lub warstwami mogą być różne. Definiowane są w postaci list, oddzielnych dla poszczególnych osi współrzędnych.

Obiekt obsługujący ten typ danych to `vtkRectilinearGrid`. Dane tego typu mogą być „specjalizowaną” odmianą danych obrazowych, spotykaną np. w danych medycznych, w których pewne fragmenty są np. skanowane z większą rozdzielczością przestrzenną.

**Structured Grid** — dane zorganizowane są w regularną topologię (siatkę) i nieregularną geometrię. Topologia wyznaczana jest pośrednio poprzez określenie wymiaru danych  $(n_x, n_y, n_z)$ . Dzięki regularnej topologii, topologiczne współrzędne dowolnego punktu można łatwo wyrazić w postaci indeksów  $(i, j, k)$ : wiersza, kolumny i warstwy. Współrzędne geometryczne każdego punktu wyznaczane są bezpośrednio, na podstawie tablicy o takich samych wymiarach co dane  $(n_x, n_y, n_z)$ . Tablica ta zawiera współrzędne poszczególnych punktów danych.

Obiekt VTK obsługujący te dane to `vtkStructuredGrid`. Dane tego typu są szczególnie użyteczne w symulacjach numerycznych, zwłaszcza w mechanice płynów, symulacji przesyłu ciepła oraz innych obliczeniach wykorzystujących np. metodę elementów skończonych.

**Polygonal Data** — są to dane, które mogą być bezpośrednio wyświetlone przez kartę graficzną, tzw. podstawowe obiekty graficzne. Do danych tych zalicza się: punkty, zbiory punktów, linie, krzywe, wielokąty oraz paski trójkątów.

Obiekt `vtkPolyData` umożliwia obsługę tego typu danych. Głównym ich zastosowaniem jest modelowanie obiektów przestrzennych.

**Unstructured Grid** — są to dane o dowolnej organizacji, zarówno topologia, jak i geometria punktów danych przechowywana jest w sposób bezpośredni, w postaci jawnych obiektów. Dowolne dane można zapisać w tym typie, jednak wiąże się to ze znacznym zapotrzebowaniem na pamięć oraz znacznymi kosztami obliczeniowymi.

Obiekt `vtkUnstructuredGrid` jest implementacją danych o dowolnej organizacji. Teoretycznie, format ten może służyć do przechowywania dowolnych danych, jednak ze względu na małą wydajność oraz duże zapotrzebowanie na pamięć korzysta się z niego jedynie w ostateczności.

**Unstructured Points** — dane w postaci jedynie punktów, bez topologii oraz z nieregularną geometrią. Jest to prosty w budowie typ danych, jednak dość znaczący. Często zdarza się, że dane tego typu są danymi wejściowymi, np. zbiorem wartości uzyskanych z „punktów pomiarowych”, a celem jest wizualizacja tych danych. Wówczas na drodze obliczeń, w strumieniu wizualizacji, wyznaczana jest „nowa” topologia danych.

Tego typu dane mogą być przechowywane zarówno w `vtkPolyData`, jak i w `vtkUnstructuredGrid`.

### A.1.8 Podsumowanie

Biblioteka VTK jest obecnie bardzo szeroko wykorzystywanym narzędziem do tworzenia aplikacji naukowych. Oferuje bardzo rozbudowane i jednocześnie elastyczne narzędzia, które upraszczają tworzenie zaawansowanych programów do wizualizacji i przetwarzania danych.

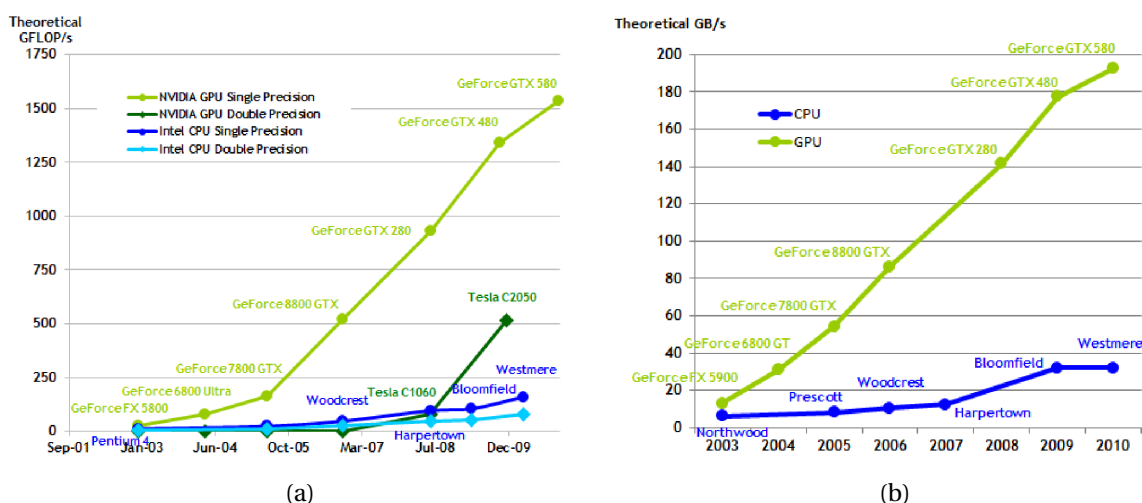
W badaniach przeprowadzonych przez autora rozprawy, biblioteka VTK posłużyła głównie do implementacji wydajnego i elastycznego modelu drzewa oskrzelowego. W implementacji wykorzystano strumieniowe możliwości przetwarzania danych — zbudowano strumień umożliwiający wygenerowanie powierzchni wirtualnego drzewa oskrzelowego, a następnie jego wydajną wizualizację. Wykorzystano wbudowane typy zbiorów danych, głównie *poly-data*, oraz możliwości sterowania procesem tworzenia obrazu poprzez użycie obiektów modelu graficznego VTK.

## A.2 Biblioteka CUDA

Karty graficzne na początku swej historii służyły jedynie do „wyświetlania” tekstu, z czasem również grafiki. Coraz bardziej zaawansowane graficzne systemy operacyjne oraz przede wszystkim gwałtowny rozwój rynku gier komputerowych doprowadził do opracowania *akceleratorów graficznych* — specjalizowanych koprocesorów umieszczanych na kartach graficznych. Realizowały one najbardziej kosztowne obliczeniowo fragmenty algorytmów generowania dwuwymiarowego obrazu, takie jak np. kopiowanie i wypełnianie kolorem, rysowanie linii, rysowanie kursora myszy itp.

Kolejnym krokiem w rozwoju były akceleratory grafiki trójwymiarowej, które w połowie lat 90. XX wieku trafiły na rynek konsumencki w relatywnie niskiej cenie. Procesor komputera dostarczał do akceleratora 3D karty graficznej opis trójwymiarowej sceny, zaś karta graficzna generowała jej obraz przy użyciu stałego i mocno ograniczonego zestawu funkcji. Stosowane algorytmy „zaszyte” były początkowo w postaci sprzętowej, z niewielkimi możliwościami sterowania. Jednak dzięki ścisłej specjalizacji oraz wysokiej optymalizacji oferowały dużą wydajność oraz odciążały główny procesor komputera. Kolejnym krokiem w ewolucji kart graficznych było opracowanie *programowalnych procesorów graficznych*, które mogły realizować dowolne algorytmy zapisane w postaci krótkich programów tzw. *shaderów*. Ze względu na możliwość programowania, dość szybko zaczęto używać procesorów graficznych do obliczeń nie związanych ściśle z generowaniem grafiki. Techniki wykorzystujące kartę graficzną do akceleracji obliczeń nazwane są akronimem GPGPU (ang. *general-purpose computing on graphics processing units*).

Współczesne karty graficzne wyposażone są w specjalizowane, wielordzeniowe procesory graficzne (ang. *Graphics Processing Unit* — *GPU*), których stopień skomplikowania oraz wydajność są znacznie większe od najwydajniejszych procesorów ogólnego zastosowania (ang. *Central Processing Unit* — *CPU*). Rysunek A.6 prezentuje historię wzrostu maksymalnej wydajności oraz przepustowości wybranych modeli kart graficznych w porównaniu z procesorami CPU firmy Intel. Opracowana przez firmę NVIDIA zunifikowana architektura procesora graficznego



Rysunek A.6: Wzrost wydajności procesorów graficznych GPU w porównaniu z procesorami CPU: (a) wzrost wydajności przetwarzania danych oraz (b) wzrost maksymalnego transferu danych (NVIDIA, 2011b)

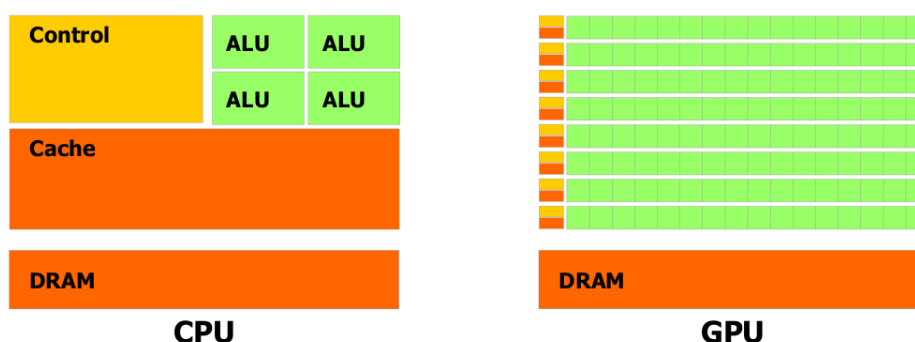
CUDA™ (ang. *Compute Unified Device Architecture*) umożliwia wykonywanie dowolnych obliczeń przez procesor karty graficznej. Architektura CUDA uzupełniona została kompletnym



środowiskiem programistycznym opartym na języku C, w skład którego wchodzi: kompilator (*nvcc*), debugger (*cuda-gdb*), profiler oraz dokumentacja interfejsu programowania aplikacji (ang. *CUDA API*).

Powstały również inne interfejsy programistyczne, które umożliwiają wykonywanie obliczeń na procesorach graficznych GPU różnych producentów. Można do nich zaliczyć otwarty i międzyplatformowy standard *OpenCL* (ang. *Open Computing Language*), którego twórcami jest konsorcjum Khronos Group oraz *DirectCompute* opracowany przez firmę Microsoft i będący częścią biblioteki DirectX. Ze względu na bardzo intensywne wsparcie ze strony firmy NVIDIA, obecnie technologia CUDA jest bardzo szeroko stosowana zarówno w rozrywce (np. akceleracja kompresji filmów), jak również w nauce. Jednak ze względu na ograniczenie możliwości użycia jedynie procesorów GPU tej firmy coraz większego znaczenia nabiera otwarty standard OpenCL, który umożliwia tworzenie aplikacji uruchamianych na heterogenicznych platformach sprzętowych, składających się z różnego rodzaju jednostek obliczeniowych (np. wielordzeniowych CPU, GPU różnych producentów oraz układów reprogramowalnych typu FPGA).

Na rysunku A.7 przedstawiono schematyczne porównanie współczesnej architektury procesora CPU oraz GPU. Rozmiary bloków funkcjonalnych procesorów są proporcjonalne do udziału w całym układzie. Można zauważyć, że podstawową różnicą między tymi architekturami jest ograniczenie w GPU do minimum części kontrolnej (*Control*) i pamięci podręcznej (*Cache*) oraz umieszczenie w jednym procesorze GPU bardzo dużej liczby rdzeni (*ALU*) wykonujących obliczenia arytmetyczno-logiczne oraz zmiennoprzecinkowe. Rdzenie GPU nie mają pełnej funkcjonalności rdzenia procesora CPU — są wyspecjalizowane do przetwarzania dużych ilości danych przez stosunkowo krótkie programy. Cechą charakterystyczną rdzeni GPU



Rysunek A.7: Schematyczne porównanie współczesnych procesorów ogólnego przeznaczenia CPU oraz procesorów graficznych GPU (NVIDIA, 2011b)

w porównaniu z rdzeniami CPU jest to, że rdzenie GPU wykonują równolegle ten sam program (*kernel*) na dużym zbiorze danych. Jest to więc technologia, która realizuje przetwarzanie równoległe na poziomie danych (ang. *Single-Program Multiple-Data* — *SPMD*) — w jednym momencie wszystkie rdzenie wykonują ten sam program. Najnowsze procesory GPU z rodziny Fermi umożliwiają również tworzenie wątków, przez to możliwe jest wykonywanie wielu programów w tym samym czasie (ang. *concurrent kernel execution*). Dzięki temu wszystkie rdzenie procesora GPU mogą być wykorzystywane w sposób bardzo efektywny, nawet podczas uruchamiania dużej ilości małych programów

Pojedynczy procesor GPU składa się z dużej liczby rdzeni, która może się wahać od kilkunastu w procesorach niskobudżetowych do kilkuset (np. 512 w procesorze karty GeForce GTX 580) w wersjach najwydajniejszych. Zestawienie podstawowych parametrów trzech głównych

rodzin procesorów graficznych firmy NVIDIA, kompatybilnych z architekturą CUDA, przedstawiono w tabeli A.1. Dane zebrane w tabeli dotyczą parametrów oraz szczytowej, teoretycznej wydajności najlepszych procesorów GPU w konkretnej architekturze. Producent oferuje również procesory z mniejszą wydajnością, które mają ograniczoną liczbę rdzeni CUDA. Procesory GPU używane są głównie w kartach graficznych przeznaczonych na rynek konsumencki (produkty serii *GeForce*) oraz profesjonalny (produkty *Quadro*). Powstała również rodzina produktów *Testa*, obejmująca specjalne wersje kart oraz autonomicznych wielordzeniowych urządzeń, które pozbawione są podzespołów umożliwiających generowanie obrazu (brak wyjść wideo), a których głównym zadaniem jest akceleracja obliczeń (ang. *High Performance Computing* — *HPC*).

Tablica A.1: Porównanie podstawowych parametrów trzech architektur procesorów graficznych GPU firmy NVIDIA zgodnych z technologią CUDA

GPU	G80	GT200	Fermi
Liczba tranzystorów	0.681 miliarda	1.4 miliarda	3.0 miliardy
Liczba rdzeni CUDA	128	240	512
Liczba multiprocessorów	16	30	16
Teoretyczna wydajność ( <i>float</i> )	0.5 TFlops	0.9 TFlops	1.5 TFlops
Teoretyczna wydajność ( <i>double</i> )	—	0.09 TFlops	0.75 TFlops
Przepustowość pamięci	100 GB/sec	140 GB/sec	190 GB/sec
Kompatybilność z wersją CUDA	1.0	1.3	2.0

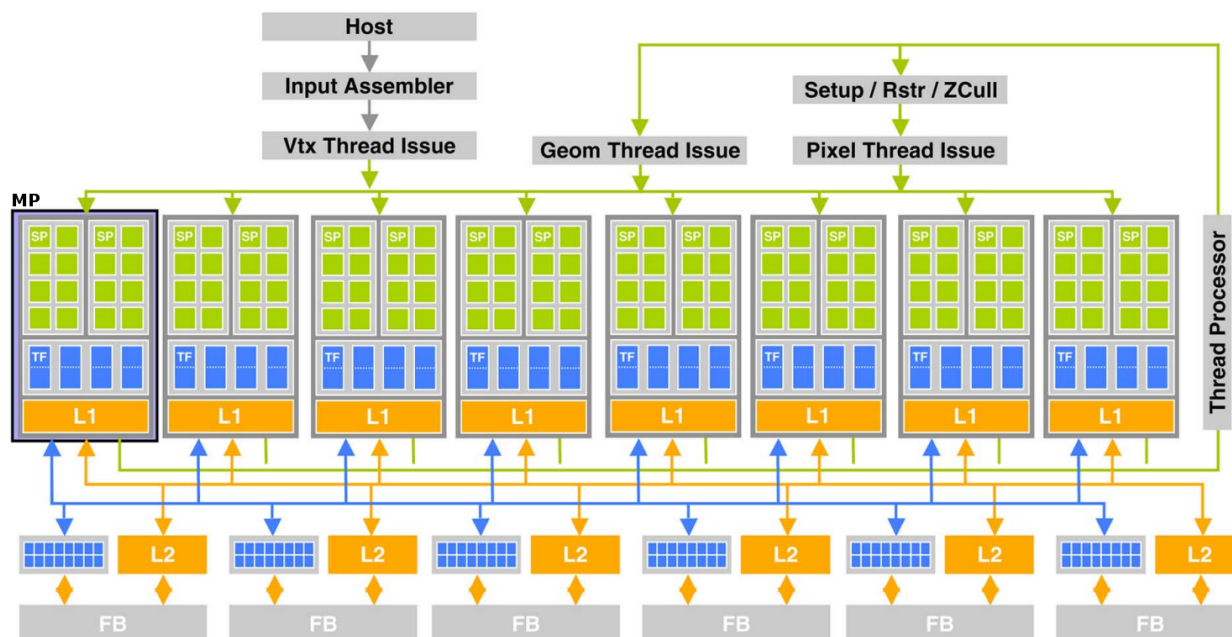
G80 – pierwsza rodzina procesorów, wprowadzona na rynek w 2006 roku; GT200 – druga generacja procesorów GPU wprowadzona w 2008 roku, która miała większą wydajność oraz wprowadzała możliwość obliczeń na liczbach podwójnej precyzji typu *double*; Fermi – procesory najnowszej generacji; teoretyczna wydajność silnie zależy od częstotliwości taktowania procesora GPU. Przytoczone wartości dotyczą kart referencyjnych.

### A.2.1 Przetwarzanie danych w GPU

Podstawy przetwarzania danych przez zunifikowane procesory graficzne zostaną omówione na przykładzie pierwszej generacji procesorów graficznych kompatybilnych z architekturą CUDA. Rysunek A.8 przedstawia schemat blokowy pierwszej generacji procesorów graficznych (architektura G80). Procesor graficzny tego typu zbudowany jest z dużej liczby *skalarnych rdzeni strumieniowych* (ang. *Stream Processor* — *SP*), które oznaczono na rysunku kolorem zielonym. Rdzenie SP są bardzo wydajnymi jednostkami skalarnymi, które umożliwiają pracę strumieniową — wyjście jednego rdzenia może stanowić wejście kolejnego. Każdy rdzeń SP umożliwia jednoczesne wykonywanie operacji MAD (mnożenie i dodawania) i MUL (mnożenie) na liczbach całkowitych oraz zmiennoprzecinkowych zgodnych ze standardem IEEE 754.

Rdzenie SP są grupowane w tzw. *multiprocessory* (ang. *Streaming Multiprocessor* — *SM*), w skład których wchodzi dodatkowo: niewielkich rozmiarów pamięć współdzielona (ang. *shared memory*), pamięć podręczna pierwszego poziomu (L1), specjalizowane jednostki odpowiedzialne za adresowanie (ang. *Texture Address unit* — *TA*) i filtrowanie tekstów (ang. *Texture Filtering unit* — *TF*) oraz specjalizowane jednostki „super-funkcjonalne” (ang. *Super Function Units* — *SFU*), umożliwiające obliczanie często występujących funkcji matematycznych, takich jak np. pierwiastek kwadratowy, odwrotność, sinus czy cosinus kąta, w kilku taktach

zegara. W kolejnych wersjach architektury GPU do multiprocesorów dodano między innymi jednostki umożliwiające obliczenia na liczbach zmiennoprzecinkowych o podwójnej precyzji (ang. *double-precision*).



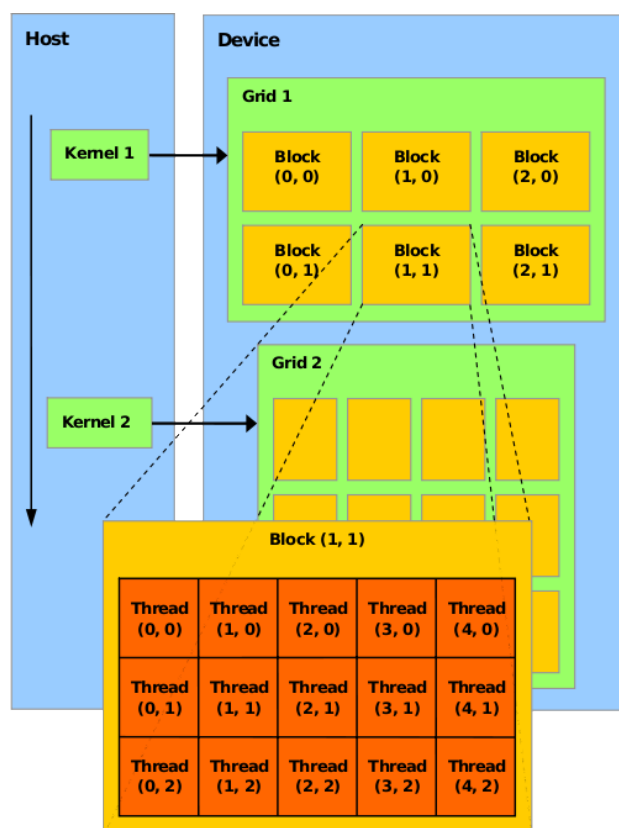
Rysunek A.8: Schemat blokowy procesora graficznego pierwszej generacji (G80), główne oznaczenia: SP – skalarnie rdzenie strumieniowe, MP – multiprocesor, TF – bloki filtrowania tekstur, L1 i L2 – pamięć podręczna, FB – pamięć obrazu (tzw. bufor ramki) (NVIDIA, 2011b)

Dane graficzne przetwarzane przez GPU mają zazwyczaj charakter wektorowy — są to np. trójwymiarowe współrzędne lub też czterokomponentowy opis koloru (RGBA). Dzięki możliwości łączenia na poziomie kodu pojedynczych procesorów w grupy, możliwe jest „doraźne tworzenie” jednostek wektorowych, które są w stanie bardzo wydajnie przetwarzać dane zarówno skalarnie, jak i wektorowe, praktycznie niezależnie od wymiaru.

Jak już wspomniano, procesor graficzny realizuje przetwarzanie równoległe na poziomie danych, co sprowadza się do tego, że w jednym momencie duża liczba rdzeni wykonuje ten sam kod programu. Na rysunku A.9 zaprezentowano w jaki sposób uruchamiane są programy (*kernels*) z poziomu CPU (*Host*) na procesorze GPU karty graficznej (*Device*).

Akceleracja obliczeń z użyciem GPU możliwa jest praktycznie w dowolnej aplikacji i polega na: przesłaniu danych do pamięci karty graficznej, a następnie wywołaniu specjalnej wersji programu (*kernel*), która zostaje uruchomiona na wielu rdzeniach GPU, realizując obliczenia w sposób równoległy. Wyniki działania *kernel* znajdują się w pamięci karty graficznej. W celu ich dalszego użycia w aplikacji należy je skopiować do pamięci operacyjnej. Biblioteka CUDA udostępnia funkcje interfejsu programistycznego, które umożliwiają: alokowanie pamięci na karcie graficznej, przesył danych, jak również umożliwiają skompilowanie programu (*kernel*) z uwzględnieniem ograniczeń konkretnej architektury GPU, a następnie jego wywołanie.

Między wątkami GPU oraz CPU istnieją istotne różnice. Przede wszystkim wątki GPU są to relatywnie krótkie fragmenty kodu, które dodatkowo posiadają bardzo małe koszty tworzenia oraz przełączania, przez co zarządzanie tysiącami wątków w procesorze odbywa się bardzo wydajnie. Cechą charakterystyczną jest również to, że cała architektura jest bardzo dobrze skalowalna na poziomie sprzętu — ten sam program (*kernel*) może być uruchomiony na różnych



Rysunek A.9: Schemat wykonywania programów (*kernels*) na wielu rdzeniach GPU (NVIDIA, 2011b)

procesorach GPU o różnej liczbie rdzeni. Czas potrzebny do wykonania obliczeń jest tym krótszy, im więcej rdzeni uczestniczy w obliczeniach.

W momencie wywołania programu (*kernel*) tworzona jest siatka (*Grid*), na którą składa się zestaw bloków (*Block*), na który z kolei składają się wątki (*Thread*) uruchamiane na konkretnych rdzeniach SP. Rozmiary siatki (liczba bloków) oraz bloku (liczba wątków) są określane w momencie uruchamiania programu *kernel* jako dodatkowe parametry. Rozmiary te dobierane są do wielkości przetwarzanych danych. Do wszystkich wątków trafia ten sam kod programu. Identyfikacja różnych wątków możliwa jest dzięki zastosowaniu specjalnych zmiennych, które dostępne są podczas wykonywania programu GPU i które mają wartość „adresu” wątku w bloku oraz bloku w siatce bloków. Wykaz zmiennych przedstawiono w tabeli A.2. Na pojedynczym rdzeniu procesora wykonywany jest więc jeden wątek *kernel*, na multiprocessorze wykonywany jest blok programów, zaś na dostępnych multiprocessorach uruchamiana jest siatka (*Grid*).

Każdy program *kernel* wykonywany jest więc przez *tablicę rdzeni*, które używają własnych identyfikatorów do adresowania danych oraz do kontroli wykonywania programu. Wymiary tablicy rdzeni określane są podczas wywoływania głównej funkcji *kernela* i dobierane są zazwyczaj do wymiarów przetwarzanych danych w taki sposób, by jeden rdzeń GPU przetwarzał jedną „porcję” danych, np. mnożył dwie konkretne wartości dwóch wektorów. Zmienne wbudowane umożliwiają zaadresowanie danych — każdy wątek może obliczyć swój własny indeks, który zostaje następnie użyty jako wskaźnik do danych. W przypadku danych jednowymiaro-

Tablica A.2: Wykaz zmiennych wbudowanych używanych do identyfikacji wątku w CUDA

Zmienna	Opis
<code>gridDim</code>	wymiary siatki, liczba bloków wątków w siatce
<code>blockDim</code>	wymiar bloku, liczba wątków w bloku
<code>blockIdx</code>	indeks bloku w siatce ( <i>grid</i> ), jedno- lub dwuwymiarowy
<code>threadIdx</code>	indeks wątku w bloku, jedno- dwu- lub trójwymiarowy

Wszystkie zmienne są typu *dim3*, który jest wbudowanym typem danych używanym w CUDA — trójelementowym wektorem liczb całkowitych dodatnich. Do elementów wektora można się odwołać przez składowe: *x*, *y* oraz *z*.

wych, indeks wątku obliczany jest na podstawie zmiennych wbudowanych z zależności (A.1):

$$\text{int } ThreadId = threadIdx.x + blockIdx.x * blockDim.x. \quad (\text{A.1})$$

Podsumowując, program *kernel* przetwarzający dane musi być napisany w taki sposób, by „umiał” zaadresować dane przeznaczone tylko dla niego, przetworzyć je, a następnie zwrócić wyniki w taki sposób, by nie zostały one nadpisane przez setki innych, działających w tym samym czasie kopii tego samego programu. Ponieważ dane są „wspólnym dobrem” wielkiej liczby rdzeni procesora graficznego, zastosowano wielopoziomową architekturę pamięci — użyto kilku rodzaj pamięci o różnych rozmiarach, wydajności oraz prawach dostępu. Przestrzeń adresowa widziana z poziomu rdzenia jest różna dla różnych typów zmiennych. Dzięki temu zwiększono wydajność przetwarzania danych, np. poprzez ograniczenie niesynchronizowanych dostępu do pamięci przez wiele rdzeni. Zastosowane w GPU bardzo szerokie szyny danych służą do przesyłu jednocześnie wielu bajtów danych. Dzięki temu, możliwe jest efektywne kopiowanie danych z pamięci globalnej do pamięci *on-chip* GPU, do której bezpośredni dostęp mają rdzenie multiprocesorów.

## A.2.2 Organizacja pamięci

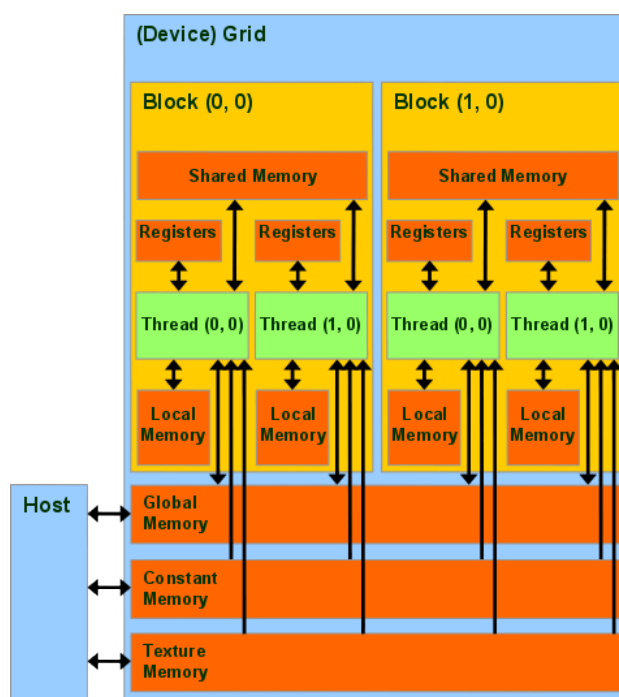
Na rysunku A.10 przedstawiono schemat wymiany danych między różnymi rodzajami pamięci, które są dostępne podczas wykonywania programu na GPU. Przestrzeń adresowa CPU (*Host*) oraz karty graficznej (*Device*) są rozdzielone. Dodatkowo, na samej karcie graficznej można wyróżnić następujące rodzaje pamięci:

- rejestry (ang. *registers*): pamięć typu *on-chip* (fizycznie w rdzeniu procesora) dostępna jedynie dla rdzeni (SP), nieadresowana przez program wątku i przeznaczona do przechowywania zmiennych,
- pamięć lokalna (ang. *local memory*): pamięć *off-chip* zlokalizowana poza rdzeniem GPU w obszarze pamięci DDR karty, dostępna jedynie przez SP na czas wykonywania wątku,
- pamięć współdzielona (ang. *shared memory*): pamięć *on-chip*, wspólna dla pojedynczego multiprocesora (bloku rdzeni) z możliwością zapisu i odczytu, o niewielkich rozmiarach (w G80 ma wielkość 16KB), szybka jak rejestr jeśli nie występują konflikty dostępu oraz dostępna tak długo, jak długo wykonywany jest blok wątków,
- pamięć globalna urządzenia (ang. *global memory*): pamięć typu *off-chip* zlokalizowana w obszarze pamięci DDR karty, znacznych rozmiarów (rzędu gigabajtów), dostępna do odczytu i zapisu dla wszystkich wątków GPU (siatek wątków) działających na rdzeniach

SP oraz dla programu hosta (CPU), dostępna od momentu alokacji do zwolnienia pamięci,

- pamięć dla wartości stałych (ang. *constant memory*): pamięć tego samego typu co pamięć globalna, z tą różnicą, że może być jedynie odczytywana przez wątki,
- pamięć tekstur (ang. *texture memory*): pamięć tego samego typu co pamięć globalna, jednak przeznaczona do przechowywania danych graficznych, do których mają dostęp specjalizowane jednostki obsługujące tekstury, możliwy jest jedynie odczyt danych przez wątki.

Wszystkie wymienione rodzaje pamięci posiadają oddzielne przestrzenie adresowe oraz różne zakresy ważności danych, na które trzeba zwracać uwagę podczas projektowania programów. Czwarta wersja biblioteki CUDA umożliwia traktowanie różnych obszarów pamięci wewnętrznej kary graficznej o architekturze Fermi jak obszaru ciągłego (ang. *Unified Address Space*).



Rysunek A.10: Schemat możliwych dostępu do pamięci karty graficznej (opis w tekście)(NVIDIA, 2011b)

W celu przyspieszenia transferu danych między hostem a kartą (CPU ↔ GPU), wykorzystuje się alokowanie pamięci w specjalnym trybie *page-locked*. System operacyjny gwarantuje, że taka pamięć zawsze będzie dostępna w pamięci RAM, nigdy nie zostanie odłożona np. na dysk. Pamięć taką alokuje się po stronie hosta i używa do kopiowania danych do/z karty graficznej. Operacje kopiowania są bardzo wydajne. Dodatkowo, w nowszym sprzęcie (kompatybilnym minimum z CUDA 2.2) możliwy jest dostęp z poziomu kernela do pamięci *page-locked* hosta. Alokowanie dużych obszarów w trybie *page-locked* może wpływać niekorzystnie na stabilność i wydajność całego systemu operacyjnego.

Tablica A.3: Wykaz słów kluczowych stosowanych w programach CUDA

Słowo kluczowe	Opis
<code>__device__</code>	zmienna w pamięci urządzenia (Device) lub funkcja uruchamiana na procesorze GPU,
<code>__host__</code>	funkcja uruchamiana na procesorze CPU (Host),
<code>__constant__</code>	zmienna w pamięci tylko do odczytu urządzenia (Device),
<code>__shared__</code>	zmienna w pamięci wspólnej multiprocesora
<code>__global__</code>	deklaracja głównej funkcji kernela, wymusza specjalne parametry wywołania funkcji: funkcja <code>&lt;&lt;&lt; Dg, Db, Ns &gt;&gt;&gt;</code> (parametry)

Dg – wektor *dim3* określający liczbę grup procesorów (Block) w siatce (Grid), Db – wektor *dim3* określający liczbę wątków (Thread) w bloku (Block), Ns – opcjonalny parametr określający rozmiar dodatkowej pamięci współdzielonej (shared memory) dynamicznie alokowanej dla bloku procesorów.

### A.2.3 Interfejs programisty

Podstawowym założeniem twórców platformy CUDA, obejmującej procesor graficzny (CUDA GPU) oraz środowisko programistyczne (CUDA API), było udostępnienie programistom szybkich i wygodnych narzędzi do tworzenia programów wykorzystujących możliwości obliczeniowe GPU (NVIDIA, 2011b). Wykorzystany w CUDA interfejs programistyczny (CUDA API) opiera się na języku C, który został uzupełniony dodatkowymi dyrektywami oraz słowami kluczowymi. Poza interfejsem CUDA dostępne są również inne interfejsy programistyczne, które umożliwiają uruchomienie na GPU programów napisanych w językach Fortran, Python, OpenCL czy też DirectCompute. W najnowszej, czwartej wersji biblioteki, przebudowano architekturę procesora graficznego Fermi w taki sposób, że możliwe jest użycie kodu zorientowanego obiektowo (ang. *object-oriented*) — kod kernela CUDA może być napisany z wykorzystaniem języka C++ (NVIDIA, 2011b).

Ponieważ języki wysokiego poziomu, takie jak C lub C++, nie posiadają możliwości zaawansowanego zarządzania pamięcią niezbędnego do adresowania różnych obszarów pamięci procesora GPU, niezbędne było uzupełnienie języka o taką funkcjonalność. Osiągnięto ją poprzez wprowadzenie słów kluczowych (tabela A.3), których użycie jednoznacznie definiuje docelowe miejsce w konkretnej przestrzeni adresowej oraz sposób uruchamiania funkcji.

Kod programu (*kernel*), który ma być wykonany w rdzeniu SP, nie może odwoływać się do pamięci hosta oraz nie może zwracać żadnej wartości (dozwolony jest jedynie *void*). Dodatkowo kod taki nie może być wywoływany rekurencyjnie, nie może używać zmiennych statycznych oraz lista argumentów nie może być zmienna. Argumenty funkcji są automatycznie kopiowane z hosta do GPU.

Biblioteka CUDA udostępnia dwie wersje interfejsów programistycznych (API):

1. niskopoziomą wersję *CUDA driver API*, dostępną przez dynamiczną bibliotekę `cuda`, dającą lepszą kontrolę nad kodem oraz niezależność od języka, jednak wymagająca większej ilości kodu, wszystkie funkcje mają prefiks `cu`,
2. wysokopoziomą wersję *CUDA runtime API*, dostępną w postaci biblioteki dynamicznej `cudaart`, oferującą mniejszą kontrolę w zamian za proste zarządzanie kodem, funkcje tej wersji mają prefiks `cuda`.

Opisane powyżej interfejsy wzajemnie się wykluczają — w jednej aplikacji można użyć funkcji tylko jednej z nich.



Dobór całkowitej liczby używanych rdzeni w blokach oraz bloków w siatce powinien uwzględniać maksymalne wykorzystanie dostępnych zasobów obliczeniowych. W najprostszym przypadku sprowadza się to do użycia w wywołaniu funkcji tylu boków ile jest multi-processorów w konkretnej wersji GPU.

#### **A.2.4 Podsumowanie**

Współczesne karty graficzne coraz częściej służą jako wydajne koprocesory, które z powodzeniem używane są do szybkiego przetwarzania dużych zbiorów danych. Specyficzna architektura procesorów graficznych wymusza użycia dostosowanych do niej narzędzi informatycznych. Biblioteka CUDA umożliwia tworzenie oprogramowania, które jest w stanie wykorzystać ogromny potencjał drzemący w procesorach GPU. Wraz z kolejnymi wersjami biblioteki oraz rozwojem procesorów graficznych, stopień skomplikowania takiego oprogramowania staje się coraz mniejszy.

W niniejszej rozprawie biblioteki CUDA użyto do przyśpieszenia kluczowych fragmentów algorytmu śledzenia końcówki endoskopu, czyli do interpolacji danych oraz wyliczania korelacji wzajemnej obrazów.

## BIBLIOGRAFIA

- Asari, K. V., Kumar, S., Radhakrishnan, D., 1999. *A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation*. Medical Imaging, IEEE Transactions on medical Imaging, tom 18, nr 4: s. 345–354. ISSN 0278-0062. doi:10.1109/42.768843.
- Bartz, D., 2005. *Virtual endoscopy in research and clinical practice*. Computer Graphics Forum, tom 24, nr 1: s. 111–126. ISSN 1467-8659. doi:10.1111/j.1467-8659.2005.00831.x.
- Beauchemin, S. S., Barron, J. L., 1995 wrz. *The computation of optical flow*. ACM Comput. Surv., tom 27, nr 3: s. 433–466. ISSN 0360-0300. doi:10.1145/212094.212141.
- Blinn, J. F., 1977 lip. *Models of light reflection for computer synthesized pictures*. SIGGRAPH Comput. Graph., tom 11, nr 2: s. 192–198. ISSN 0097-8930. doi:10.1145/965141.563893.
- Bouguet, J. Y., 2008. *Camera calibration toolbox for Matlab*.
- Bourke, P., March 1987. *Texture mapping schemes in common usage*. [http://paulbourke.net/texture\\_colour/texturemap/](http://paulbourke.net/texture_colour/texturemap/).
- Bradski, G. R., Davis, J. W., 2002 lip. *Motion segmentation and pose recognition with motion history gradients*. Mach. Vision Appl., tom 13, nr 3: s. 174–184. ISSN 0932-8092. doi:10.1007/s001380100064.
- Bricault, I., Ferretti, G., Cinquin, P., 1998. *Registration of real and CT-derived virtual bronchoscopic images to assist transbronchial biopsy*. Medical Imaging, IEEE Transactions on Medical Imaging, tom 17, nr 5: s. 703–714. ISSN 0278-0062. doi:10.1109/42.736022.
- Briechle, K., Hanebeck, U., 2001. *Template matching using fast normalized cross correlation*. W *Aerospace/Defense Sensing, Simulation, and Controls*, s. 95–102. International Society for Optics and Photonics.
- Bronsztejn, I., Siemiendajew, K., 1973. *Matematyka poradnik encyklopedyczny: Wyd. 14*. Państwowe Wydawnictwo Naukowe.
- Bułat, J., Duda, K., Duplaga, M., Frączek, R., Skalski, A., Socha, M., Turcza, P., Zieliński, T., 2007. *Data processing tasks in wireless gi endoscopy: Image-based capsule localization & navigation and video compression*. W *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, s. 2815–2818. IEEE.
- Bułat, J., Duda, K., Skalski, A., Socha, M., Turcza, P., Twardowski, T., Zieliński, T., Duplaga, M., 2006 IX. *Image and video processing tasks in computer aided medical interventions on the example of transbronchial biopsy*. W „*New trends in audio and video*”: XI symposium AES, s. 34–42.

- Bułat, J., Duda, K., Socha, M., Turcza, P., Zieliński, T., Duplaga, M., 2008. *Computational tasks in bronchoscope navigation during computer-assisted transbronchial biopsy*. W *Proceedings of the 8th international conference on Computational Science, Part III, ICCS '08*, s. 178–187. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-540-69388-8. doi:10.1007/978-3-540-69389-5\_21.
- Bułat, J., Duda, K., Socha, M., Turcza, P., Zieliński, T., Duplaga, M., 2010. *Computational tasks in computer-assisted transbronchial biopsy*. *Future Generation Computer Systems*, tom 26, nr 3: s. 455–461.
- Chung, A. J., Edwards, P. J., Deligianni, F., Yang, G.-Z., 2004. *Freehand cocalibration of optical and electromagnetic tracker for navigated bronchoscopy*. W *The Second International Workshop on Medical Imaging and Augmented Reality (MIAR 2004)*, s. 320–328.
- Cierniak, R., 2005. *Tomografia komputerowa. Budowa urządzeń CT. Algorytmy rekonstrukcyjne*. EXIT. ISBN 83-60434-00-X.
- Cook, R. L., Torrance, K. E., 1981. *A reflectance model for computer graphics*. W *Proceedings of the 8th annual conference on Computer graphics and interactive techniques, SIGGRAPH '81*, s. 307–316. ACM, New York, NY, USA. ISBN 0-89791-045-1. doi:10.1145/800224.806819.
- Cyganek, B., 2002. *Komputerowe przetwarzanie obrazów trójwymiarowych*. Problemy Współczesnej Nauki: Informatyka. Akademicka Oficyna Wydawnicza EXIT. ISBN 9788387674342.
- Daniilidis, K., 1997. *Fixation simplifies 3D motion estimation*. *Computer Vision and Image Understanding*, tom 68, nr 2: s. 158–169. ISSN 1077-3142. doi:10.1006/cviu.1997.0535.
- Deligianni, F., Chung, A., Yang, G.-Z., 2005. *Predictive camera tracking for bronchoscope simulation with CONDensation*. W *Condensation, International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI05), Palm*, s. 910–916.
- Deligianni, F., Chung, A. J., Yang, G.-Z., 2003. *pq-Space based 2D/3D registration for endoscope tracking*. W R. E. Ellis, T. M. Peters, red., *MICCAI (1)*, tom 2878 z serii *Lecture Notes in Computer Science*, s. 311–318. Springer. ISBN 3-540-20462-8.
- Dempski, K., Viale, E., 2005. *Advanced Lighting and Materials with Shaders*. Wordware Publishing Inc. ISBN 1-55622-292-0.
- Donner, C., Jensen, H. W., 2006 czer. *A spectral BSSRDF for shading human skin*. W *Rendering Techniques 2006: 17th Eurographics Workshop on Rendering*, s. 409–418.
- Duda, K., Zieliński, T., Socha, M., Twardowski, T., Duplaga, M., 2006. *Navigation in bronchial tree based on motion estimation and mutual information*. W *International Conference on Signals and Electronic Systems*, tom 2, s. 683–686.
- Duplaga, M., Bułat, J., Leszczuk, M., Socha, M., Romaniak, P., Turcza, P., 2010. *The BRONCHOVID – computer system supporting bronchoscopy laboratory*, tom 2 z serii *Advances in Intelligent and Soft Computing*. Springer-Verlag.
- Duplaga, M., Socha, M., Wojciechowski, W., 2006. *The impact of the support of the bronchial needle aspiration with interactive virtual bronchoscopy application on the effectiveness of the procedure*. W *Biomedical engineering, proceedings of international conference*, s. 17–21. Kauno Technologijos Universitetas.

- Duplaga, M., Socha, M., Wojciechowski, W., Soja, J., Tomaszewska, R., Urbanik, A., Nizankowska-Mogilnicka, E., Sladek, K., 2008. *Comparison of the effectiveness of the transbronchial needle aspiration supported with the interactive virtual bronchoscopy application or with non-real time endobronchial ultrasonography in patients suspected of lung cancer*. European Respiratory Society Annual Congress Berlin Germany October 48, s. [P3396].
- Gildea, T. R., Mazzone, P. J., Karnak, D., Meziane, M., Mehta, A. C., 2006. *Electromagnetic navigation diagnostic bronchoscopy: A prospective study*. Am. J. Respir. Crit. Care Med., tom 174, nr 9: s. 982–989. doi:10.1164/rccm.200603-344OC.
- Gray, H., 1918. *Anatomy of the human body*. Philadelphia: Lea and Febiger. ISBN 1-58734-102-6.
- Haas, A. R., Vachani, A., Sterman, D. H., 2010. *Advances in diagnostic bronchoscopy*. American Journal of Respiratory and Critical Care Medicine. doi:10.1164/rccm.201002-0186CI.
- Hansen, G., Douglass, R., Zardecki, A., 2005. *Mesh Enhancement: Selected Elliptic Methods, Foundations And Applications*. Imperial College Press. ISBN 9781860944871.
- Haponik, E., Cappellari, J., Chin, R., Adair, N., Lykens, M., Alford, P., Bowton, D., 1995. *Education and experience improve transbronchial needle aspiration performance*. Am J Respir Crit Care Med, tom 151, nr 6: s. 1998–2002. ISSN 1073-449X.
- Harris, M., 2008. *Optimizing Parallel Reduction in CUDA*. nVidia.
- Hartley, R. I., Zisserman, A., 2004. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2. wyd.
- Helferty, J., Higgins, W., 2001 X. *Technique for registering 3D virtual CT images to endoscopic video*. W *Image Processing, 2001. Proceedings. 2001 International Conference on*, tom 2, s. 893–896. doi:10.1109/ICIP.2001.958638.
- Helferty, J., Higgins, W., 2002. *Combined endoscopic video tracking and virtual 3D CT registration for surgical guidance*. W *Image Processing, 2002. Proceedings. 2002 International Conference on*, tom 2, s. II-961 – II-964. ISSN 1522-4880. doi:10.1109/ICIP.2002.1040112.
- Helferty, J., Zhang, C., McLennan, G., Higgins, W., 2001 VII. *Videoendoscopic distortion correction and its application to virtual guidance of endoscopy*. Medical Imaging, IEEE Transactions on, tom 20, nr 7: s. 605–617. ISSN 0278-0062. doi:10.1109/42.932745.
- Herth, F., Ernst, A., Becker, H., 2002. *Endobronchial ultrasound-guided transbronchial lung biopsy in solitary pulmonary nodules and peripheral lesions*. European Respiratory Journal, tom 20, nr 4: s. 972–974. doi:10.1183/09031936.02.00032001.
- Higgins, W. E., Helferty, J. P., Padfield, D. R., 2003. *Integrated bronchoscopic video tracking and 3D CT registration for virtual bronchoscopy*. Proceedings of SPIE, tom 5031: s. 80–89.
- Hii, A., Hann, C., Chase, J., Van Houten, E., 2006. *Fast normalized cross correlation for motion tracking using basis functions*. Computer methods and programs in biomedicine, tom 82, nr 2: s. 144–156.
- Horn, B. K. P., Schunck, B. G., 1981. *Determining optical flow*. ARTIFICIAL INTELLIGENCE, tom 17: s. 185–203.

- Hounsfield, G. N., 1973. *Computerized transverse axial scanning (tomography): Part 1. description of system*. Br J Radiol, tom 46, nr 552: s. 1016–1022. doi:10.1259/0007-1285-46-552-1016.
- Kak, A. C., Slaney, M., 2001. *Principles of Computerized Tomographic Imaging*. Society of Industrial and Applied Mathematics.
- Kitware Inc., 2004. *The VTK User's Guide*. Kitware Inc.
- Kitware Inc., 2011. *Shader in VTK*. [http://www.itk.org/Wiki/Shader\\_In\\_VTK](http://www.itk.org/Wiki/Shader_In_VTK).
- Lardinois, D., Brack, T., Gaspert, A., Spahr, T., Schneiter, D., Steinert, H., Weder, W., 2003. *Bronchoscopic radioisotope injection for sentinel lymph-node mapping in potentially resectable non-small-cell lung cancer*. Eur J Cardiothorac Surg, tom 23, nr 5: s. 824–7.
- Lardinois, D., De Leyn, P., Van Schil, P., Porta, R. R., Waller, D., Passlick, B., Zielinski, M., Lerut, T., Weder, W., 2006. *ESTS guidelines for intraoperative lymph node staging in non-small cell lung cancer*. Eur J Cardiothorac Surg, tom 30, nr 5: s. 787–92. ISSN 1010-7940.
- Lewis, J. P., 1995. *Fast normalized cross-correlation*.
- Lewis, R. R., 1994. *Making shaders more physically plausible*. W In *Fourth Eurographics Workshop on Rendering*, s. 47–62.
- Liu, Y., So, R., Zhang, C., 2003. *Modeling the bifurcating flow in an asymmetric human lung airway*. Journal of Biomechanics, tom 36, nr 7: s. 951–959. ISSN 0021-9290. doi:10.1016/S0021-9290(03)00064-2.
- Lorensen, W. E., Cline, H. E., 1987. *Marching cubes: A high resolution 3D surface construction algorithm*. SIGGRAPH Comput. Graph., tom 21, nr 4: s. 163–169. ISSN 0097-8930. doi:<http://doi.acm.org/10.1145/37402.37422>.
- Lucas, B. D., Kanade, T., 1981 April. *An iterative image registration technique with an application to stereo vision (IJCAI)*. W *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, s. 674–679.
- Martin, K., Hoffman, B., 2003 01. *Mastering CMake: A Cross-Platform Build System*. Kitware Inc.
- Martinez-Monge, R., Fernandes, P. S., Gupta, N., Gahbauer, R., 1999. *Cross-sectional nodal atlas: A tool for the definition of clinical target volumes in three-dimensional radiation therapy planning*. Radiology, tom 211, nr 3: s. 815–828.
- Marzat, J., 2009. *Lucas & kanade pyramidal refined optical flow implementation*. <http://www.mathworks.com/matlabcentral/fileexchange/22950-lucas-kanade-pyramidal-refined-optical-flow-implementation>.
- Mori, K., Deguchi, D., Akiyama, K., Kitasaka, T., Maurer Jr., C. R., Suenaga, Y., Takabatake, H., Mori, M., Natori, H., 2005. *Hybrid bronchoscope tracking using a magnetic tracking sensor and image registration*. W J. Duncan, G. Gerig, red., *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005*, tom 3750 z serii *Lecture Notes in Computer Science*, s. 543–550. Springer Berlin Heidelberg. ISBN 978-3-540-29326-2. doi:10.1007/11566489\_67.

- Mori, K., Deguchi, D., Sugiyama, J., Suenaga, Y., Toriwaki, J.-i., Maurer Jr., C. R., Takabatake, H., Natori, H., 2002. *Tracking of a bronchoscope using epipolar geometry analysis and intensity-based image registration of real and virtual endoscopic images*. Medical Image Analysis, tom 6, nr 3: s. 321–336.
- Mori, K., Enjoji, T., Deguchi, D., Kitasaka, T., Suenaga, Y., Toriwaki, J., Takabatake, H., Natori, H., 2004. *New image similarity measures for bronchoscope tracking based on image registration between virtual and real bronchoscopic images*. s. 165–176. doi:10.1117/12.536385.
- Mountain, C. F., Dresler, C. M., 1997. *Regional lymph node classification for lung cancer staging*. Chest, tom 111(6): s. 1486–7.
- Munch, J., 2010 february. *Electromagnetic navigation biopsy for peripheral lung lesion diagnosis*. OncoLog, tom 55, nr 2.
- Murray, D. W., Buxton, B. F., 1987 III. *Scene segmentation from visual motion using global optimization*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, tom PAMI-9, nr 2: s. 220 –228. ISSN 0162-8828. doi:10.1109/TPAMI.1987.4767896.
- Nagao, J., Mori, K., Enjouji, T., Deguchi, D., Kitasaka, T., Suenaga, Y., Hasegawa, J.-i., Toriwaki, J.-i., Takabatake, H., Natori, H., 2004. *Fast and accurate bronchoscope tracking using image registration and motion prediction*. W C. Barillot, D. R. Haynor, P. Hellier, red., *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2004*, tom 3217 z serii *Lecture Notes in Computer Science*, s. 551–558. Springer Berlin Heidelberg. ISBN 978-3-540-22977-3. doi: 10.1007/978-3-540-30136-3\_68.
- NEMA, 2009. *Digital imaging and communications in medicine (DICOM) part 14: Grayscale standard display function*. Rap. tech., National Electrical Manufacturers Association.
- Nicodemus, F. E., 1965 Jul. *Directional reflectance and emissivity of an opaque surface*. Appl. Opt., tom 4, nr 7: s. 767–773. doi:10.1364/AO.4.000767.
- NVIDIA, 2011a. „*CUDA C best practices guide version 4.0*”. Rap. tech., NVIDIA Corporation.
- NVIDIA, 2011b. „*CUDA C programming guide version 4.0*”. Rap. tech., NVIDIA Corporation.
- NVIDIA, 2012. „*NVIDIA CUDA Fast Fourier Transform (FFT) library*”. <http://docs.nvidia.com/cuda/cufft/index.html>.
- Olimpus, 2011. *Materiały reklamowe*. [http://www.olympusamerica.com/msg\\_section/msg\\_endoscopy\\_video\\_bronchoscopes.asp](http://www.olympusamerica.com/msg_section/msg_endoscopy_video_bronchoscopes.asp).
- Oren, M., Nayar, S. K., 1994. *Generalization of lambert's reflectance model*. W *In SIGGRAPH 94*, s. 239–246. ACM Press.
- Pirożyński, M., 2011. *Bronchofiberoskopia*. Wyd. 2. Alfa Medica Press. ISBN 978-83-7522-069-8.
- Preim, B., Bartz, D., 2007. *Visualization in Medicine: Theory, Algorithms, and Applications*. The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISBN 0123705967.
- Randima, F., Kilgard, M. J., 2003. *Język Cg. Programowanie grafiki w czasie rzeczywistym*.

- Schlick, C., 1994. *An inexpensive BRDF model for physically-based rendering*. Computer Graphics Forum, tom 13, nr 3: s. 233–246. ISSN 1467-8659. doi:10.1111/1467-8659.1330233.
- Schroeder, W., Martin, K., Lorensen, B., 2004. *The Visualization Toolkit, Third Edition*. Kitware Inc. ISBN 1930934122.
- Schroeder, W., Zarge, J., Lorensen, W., 1992. *Decimation of triangle meshes*. W *ACM SIGGRAPH Computer Graphics*, tom 26, s. 65–70. ACM.
- Shah, S., Aggarwal, J., 1996. *Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation*. Pattern Recognition, tom 29, nr 11: s. 1775–1788. ISSN 0031-3203. doi:10.1016/0031-3203(96)00038-6.
- Sherbondy, A. J., Kiraly, A. P., Austin, A. L., Helferty, J. P., Wan, S.-Y., Turlington, J. Z., Yang, T., Zhang, C., Hoffman, E. A., McLennan, G., Higgins, W. E., 2000. *Virtual bronchoscopic approach for combining 3D CT and endoscopic video*. Proceedings of SPIE, tom 3978: s. 104–116. doi:10.1117/12.383389.
- Shields, T. W., LoCicero, J., Ponn, R. B., Rusch, V. W., 2004. *General Thoracic Surgery*, tom 2. Lippincott Williams & Wilkins, 6. wyd. ISBN 078173889X.
- Siemens, 2011. *Somatom definition flash*. [http://www.siemens.pl/home/siemens\\_med\\_tk\\_Somatom\\_Sensation.asp](http://www.siemens.pl/home/siemens_med_tk_Somatom_Sensation.asp).
- Siemens AG Medical Solutions, 2011. *Computed tomography, its history and technology*. [http://www.medical.siemens.com/siemens/en\\_US/gg\\_ct\\_FBAs/files/brochures/CT\\_History\\_and\\_Technology.pdf](http://www.medical.siemens.com/siemens/en_US/gg_ct_FBAs/files/brochures/CT_History_and_Technology.pdf).
- Skalski, A., Socha, M., Duplaga, M., Duda, K., Zieliński, T., 2010. *3D segmentation and visualization of mediastinal structures adjacent to tracheobronchial tree from CT data*. W E. Piętka, J. Kawa, red., *Information Technologies in Biomedicine*, tom 69 z serii *Advances in Intelligent and Soft Computing*, s. 523–534. Springer Berlin Heidelberg. ISBN 978-3-642-13104-2. doi: 10.1007/978-3-642-13105-9\_52.
- Skalski, A., Socha, M., Zieliński, T., Duplaga, M., 2011. *Virtual colonoscopy – technical aspects*. W P. Miskovitz, red., *Colonoscopy*, s. 271–290. InTech. ISBN 978-953-307-568-6. doi:10.5772/19329.
- Skalski, A., Socha, M., Zieliński, T., Duplaga, M., 2007a. *Colon cleansing for virtual colonoscopy using non-linear transfer function and morphological operations*. W *Imaging Systems and Techniques, 2007. IST'07. IEEE International Workshop on*, s. 1–5. IEEE.
- Skalski, A., Socha, M., Zieliński, T., Duplaga, M., 2007b. *CT data processing and visualization aspects of Virtual Colonoscopy*. Blood, tom 80: s. 10.
- Socha, M., 2011a. *Parametryczny model powierzchni drzewa oskrzelowego*. W *Materiały XV Sympozjum nt.: "Modelowanie i Symulacja Systemów Pomiarowych"*, Krynica, s. 101–104. ISBN 978-83-61528-24-1.
- Socha, M., 2011b. *Wizualizacja danych biomedycznych*. Wydawnictwa AGH. ISBN 978-83-7464-376-4.



- Socha, M., 2013. *Szybka metoda estymacji położenia końcówki bronchoskopu — implementacja w GPU (zgłoszony do recenzji)*. XII Sympozjum Modelowanie i Pomiary w Medycynie MPM 2013, Krynica Górská.
- Socha, M., Duda, K., Zieliński, T. P., Duplaga, M., 2005a. *Algorytmiczna korekcja zniekształceń geometrycznych kamery bronchoskopu*. W *Modelowanie i Pomiary w Medycynie MPM 2005*, Krynica, s. 219–228.
- Socha, M., Duplaga, M., 2005. *Aplikacja oparta na bibliotece VTK wspomagająca zabiegi bronchoskopowe*. BIO-ALGORITHMS AND MED-SYSTEMS, tom 1, nr 1/2: s. 191–196.
- Socha, M., Duplaga, M., Turcza, P., 2004. *Methods of bronchial tree reconstruction and camera distortion corrections for virtual endoscopic environments*. *Studies in Health Technology and Informatics*, tom 105: s. 285–295.
- Socha, M., Duplaga, M., Turcza, P., Zieliński, T. P., 2005b. *Oprogramowanie wspomagające zabieg transbronchialnej igłowej biopsji aspiracyjnej metodą wirtualnej bronchoskopii*. s. 115–122. VII Sympozjum Modelowanie i Pomiary w Medycynie MPM 2005, Krynica Górská.
- Socha, M., Zieliński, T. P., Duplaga, M., 2009. *Symulowanie charakterystycznych dla endoskopu zjawisk zniekształcających obraz*. W *Modelowanie i Pomiary w Medycynie MPM 2009*, Krynica, s. 175–178.
- Solomon, S. B., White Jr., P., Wiener, C. M., Orens, J. B., Wang, K. P., 2000. *Three-dimensional CT-guided bronchoscopy with a real-time electromagnetic position sensor: A comparison of two image registration methods*. *CHEST Journal*, tom 118, nr 6: s. 1783–1787. doi:10.1378/chest.118.6.1783.
- Stevens, A., Lowe, J., 2000. *Histologia człowieka*. Wydawnictwo Lekarskie PZWL. ISBN 8320024013.
- Tadeusiewicz, R., Korohoda, P., 1997. *Komputerowa analiza i przetwarzanie obrazów*. Społeczeństwo Globalnej Informacji. Wydawnictwo Fundacji Postępu Telekomunikacji. ISBN 9788386476152.
- Tian, T., Tomasi, C., Heeger, D., 1996 VI. *Comparison of approaches to egomotion computation*. W *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, s. 315–320. ISSN 1063-6919. doi:10.1109/CVPR.1996.517091.
- Toloza, E., Harpole, L., McCrory, D., 2003. *Noninvasive staging of non-small cell lung cancer: a review of the current evidence*. *Chest*, tom 123, nr 1 Suppl: s. 137S–146S.
- Torrance, K., Sparrow, E., 1967 wrz. *Theory for off-specular reflection from roughened surfaces*. *Journal of the Optical Society of America (JOSA)*, tom 57, nr 9: s. 1105–1114.
- Trucco, E., Verri, A., 1998. *Introductory techniques for 3-D computer vision*. Prentice Hall. ISBN 9780132611084.
- Turcza, P., 2005a. *Multiscale optimization strategy in image registration based navigation system for bronchofiberscopic procedures*. 14-th IMEKO Symposium on New technologies in Measurement and Instrumentation, s. 297–301.

- Turcza, P., 2005b. *Navigation system for bronchofiberoptic procedures based on image registration with scale adaptive image similarity measure*. European Signal Processing Conference EUSIPCO-2005.
- Turcza, P., Duplaga, M., 2004. *Navigation systems based on registration of endoscopic and CT-derived virtual images for bronchofiberoptic procedures*. W *Transformation of Healthcare with Information Technologies*, nr 105 w Studies in Health Technology and Informatics, s. 253–263. IOS Press. ISBN 978-1-58603-438-2.
- Twardowski, T., Zieliński, T., Duda, K., Socha, M., Duplaga, M., 2006 VIII. *Fast estimation of broncho-fiberscope egomotion for Ct-guided transbronchial biopsy*. W *Image Processing, 2006 IEEE International Conference on*, s. 1189 –1192. IEEE. ISSN 1522-4880. doi: 10.1109/ICIP.2006.312696.
- Viola, P., Wells, I., W.M., 1995 VI. *Alignment by maximization of mutual information*. W *Computer Vision, 1995. Proceedings., Fifth International Conference on*, s. 16–23. doi:10.1109/ICCV.1995.466930.
- Wiatr, K., 2003. *Akceleracja obliczeń w systemach wizyjnych*. WNT. ISBN 9788320428469.
- Wilson, S., Lovell, B., Chang, A., Masters, B., Brisbane 2005. *Visual odometry for quantitative bronchoscopy using optical flow*. Australian Pattern Recognition Society Workshop on Digital Image Computing: Pattern Recognition and Imaging for Medical Applications.
- Zabrodzki, J., red., 1994. *Grafika komputerowa, metody i narzędzia*. WNT.
- Zieliński, T., Bułat, J., Socha, M., Duplaga, M., 2006. *Estymacja parametrów ruchu kamery bronchofiberoskopu metodą geometrii epipolarnej i przepływu optycznego z wykorzystaniem szybkiej zespolonej transformacji falkowej*. s. 127–132. *Pomiary Automatyka Kontrola*, nr 5 bis.
- Zieliński, T., Duplaga, M., Bułat, J., Duda, K., Skalski, A., Socha, M., Turcza, P., 2007. *Model algorytmiczny komputerowego systemu nawigacyjnego do wspomagania zabiegów igłowej biopsji przezoskrzelowej metodą wirtualnej bronchoskopii*. tom 53, s. 87–94. *Pomiary Automatyka Kontrola*.
- Zieliński, T. P., 2007. *Cyfrowe przetwarzanie sygnałów: od teorii do zastosowań*. Wydawnictwa Komunikacji i Łączności. ISBN 978-83-206-1640-8.

## SPIS RYSUNKÓW

2.1	Budowa układu oddechowego człowieka (autor grafiki: Mariana Ruiz Villarreal, wikipedia) . . . . .	7
2.2	Rozmieszczenie węzłów chłonnych i innych tkanek w okolicach drzewa oskrzelowego: (a) opracowane przez Graya (Gray, 1918) oraz (b) współczesna klasyfikacja Mountain/Dreslera węzłów chłonnych (Mountain i Dresler, 1997) . . . . .	10
2.3	Obrazy CT klatki piersiowej z zaznaczonymi węzłami chłonnymi (Martinez-Monge i inni, 1999) . . . . .	11
2.4	Rozmieszczenie przekroi CT z rysunku 2.3 (Martinez-Monge i inni, 1999) . . . . .	12
2.5	Przykład endoskopu sztywnego wraz z osprzętem: okularom optycznym, oświetleniem oraz szczypcami (Shields i inni, 2004) . . . . .	12
2.6	Bronchofiberoskop z okularom optycznym firmy Olympus (Olimpus, 2011) . . . . .	13
2.7	Zabieg bronchofiberoskopii wykonywany bronchofiberoskopem optycznym (a), wideobronchofiberoskopem wyposażonym w kamerę wideo (b) oraz przykłady obrazów drzewa oskrzelowego (c) obserwowane przez lekarza podczas bronchoskopii . . . . .	13
2.8	Przykład użycia bronchofiberoskopu do: usunięcia ciała obcego, pobrania za pomocą szczoteczki próbek tkanki i wydzielin, pobrania wycinka ściany oskrzela oraz wykonania <i>biopsji aspiracyjnej cienkoigłowej</i> (BAC) (Olimpus, 2011) . . . . .	14
2.9	Ugięta końcówka bronchofiberoskopu (a) oraz powiększony widok czoła bronchofiberoskopu (b), na którym zaznaczono: 1 – kanał roboczy, 2 – soczewkę układu optycznego oraz 3 – światłowodowe oświetlenie . . . . .	14
2.10	Schemat działania spiralnej tomografii komputerowej: a) wzajemne położenie pacjenta, źródła promieniowania i detektorów oraz b) spiralna ścieżka, po której porusza się zespół źródło-detektory względem pacjenta (Siemens AG Medical Solutions, 2011) . . . . .	16
2.11	Skala Hounsfield: a) przykładowe zakresy wartości HU tkanek w szerokim zakresie wartości oraz b) wybrane, niskie wartości HU (Siemens AG Medical Solutions, 2011) . . . . .	18
2.12	Fragment klatki piersiowej przedstawiony dla różnych nastaw parametrów okna $W$ i zakresu $C$ : (a) odcienie szarości pokrywają cały zakres danych ( $C = 217, W = 2482$ ), (b) szeroki zakres wartości prezentujący tkankę płuc ( $C = -600, W = 1200$ ) oraz (c) wąski zakres prezentowanych wartości HU ( $C = 40, W = 300$ ) dobrany do wizualizacji węzłów chłonnych . . . . .	18
2.13	Sondy ultrasonograficzne stosowane podczas zabiegów bronchoskopii: (a) radialna sonda ultrasonograficzna wprowadzana przez kanał roboczy i (b) przykładowy obraz przez nią uzyskiwany oraz (c) głowica ultrasonograficzna zintegrowana z bronchofiberoskopem i (d) obraz przez nią uzyskiwany w czasie biopsji (Haas i inni, 2010) . . . . .	20

2.14	Rekonstrukcja powierzchni drzewa oskrzelowego polega na analizie serii skanów z tomografii komputerowej (a), w wyniku której powstaje przestrzenna siatka (b), która jest następnie wizualizowana (c) (Bułat i inni, 2010) . . . . .	22
3.1	Elementy opracowanego wirtualnego fantomu drzewa oskrzelowego i endoskopu, umożliwiające generowanie obrazów wirtualnej bronchoskopii . . . . .	28
3.2	Kolejne etapy modelowania kształtu drzewa oskrzelowego: (a) szkielet drzewa oskrzelowego, (b) pogrubienie węzłów, (c) modelowanie łuków rozwidleń, (d) schematyczny wynik końcowy oraz (e) przykład użycia ujemnej średnicy do modelowania przewężeń . . . . .	30
3.3	Konstrukcja łuku o zadanym promieniu $r$ , wygładzającego rozgałęzienie wyznaczone przez punkty $p_1 - p_2 - p_3$ : (a) na płaszczyźnie oraz (b) w przestrzeni, z zaznaczonymi płaszczyznami prostopadłymi do ramion łuku . . . . .	31
3.4	Przykład wygładzania rozgałęzienia przez wpisanie łuku . . . . .	35
3.5	Przykład dwuwymiarowego przekroju przez pole skalarne wygenerowane podczas pogrubiania szkieletu . . . . .	35
3.6	Wizualizacja pola skalarne wygenerowanego na podstawie szkieletu drzewa oskrzelowego . . . . .	36
3.7	Wyniki rekonstrukcji modelu powierzchni drzewa oskrzelowego: siatka trójkątów opisująca powierzchnię oraz oświetlona powierzchnia . . . . .	37
3.8	Dwukierunkowa funkcja rozkładu odbicia (BRDF) jako funkcja kątów padania światła $(\theta_L, \phi_L)$ oraz kątów obserwacji $(\theta_V, \phi_V)$ . . . . .	39
3.9	Schemat powstawania odbicia od nieidealnie równej powierzchni: (a) model odbicia rozproszonego (ang. <i>diffuse</i> ), (b) model odbicia zwierciadlanego (ang. <i>specular</i> ), (c) suma modeli odbicia rozproszonego i zwierciadlanego oraz (d) przykład rzeczywistego odbicia . . . . .	41
3.10	Wektory używane do wyznaczenia współczynników modelu oświetlenia Phong: $N$ - wektor normalny, $L$ - wektor określający kierunek padania światła, $R$ - promień odbity, $V$ - kierunek do obserwatora . . . . .	42
3.11	Porównanie modeli oświetlenia: (a) model rozproszenia lambertowskiego – cieniowanie Phong, (b) model Oren-Nayara, (c) model Oren-Nayara z dodanymi odbiciami zwierciadlanymi, (d) zjawisko Fresnela w modelu Cook-Torrance’a, (e) światło rozproszone oraz efekt Fresnela modelu Cook-Torrance’a, (f) pełen model Cook-Torrance’a obejmujący światło rozproszone, odbicie zwierciadlane oraz zjawisko Fresnela . . . . .	44
3.12	Trzy rodzaje interpolacji (cieniowania) zastosowane podczas obliczania koloru powierzchni sfery . . . . .	45
3.13	Zasada działania teksturowania przez mapowanie UV . . . . .	46
3.14	Przykład rozwinięcia siatki (a) wygenerowanej dla modelu drzewa oskrzelowego, (b) tekstura opracowana na jej podstawie oraz (c) obraz wirtualnego drzewa oskrzelowego z nałożoną teksturą . . . . .	47
3.15	Przykłady mapowania tekstury na trzy rodzaje przestrzennych figur (sferę, sześcian oraz walec) metodą rzutowania płaszczyzny (pierwsza kolumna) walca (druga kolumna) oraz sfery (trzecia kolumna) (Bourke, March 1987) . . . . .	48
3.16	Przykład użycia: (a) mapowania parametrycznego oraz (b) tekstury proceduralnej (szum Perlina) . . . . .	49

3.17	Przykład zniekształceń wprowadzanych przez tor optyczny wideobronchoskopu: (a) płaska powierzchnia z równomiernie rozmieszczonymi kopkami, (b) i (c) płaska powierzchnia z szachownicą o różnej gęstości . . . . .	51
3.18	Wyniki korekcji zniekształceń: (a) obraz z endoskopu, (b) obraz skorygowany metodą dedykowaną i (c) uniwersalną oraz (d) porównanie pozycji środków kół (— metoda dedykowana, $\cdots$ metoda uniwersalna) . . . . .	54
3.19	Porównanie wartości błędu średniokwadratowego dopasowania linii do prostych: (a) dla wierszy oraz (b) kolumn . . . . .	55
3.20	Przykłady rozmycia rzeczywistego obrazu z endoskopu spowodowane: (a) skończoną głębią ostrości oraz (b) ruchem końcówki . . . . .	56
3.21	Symulowane zniekształcenia obrazu: (a) obraz oryginalny, (b) zniekształcenia beczkowe oraz (c) poduszkowe . . . . .	56
3.22	Wyniki symulacji rozmycia obrazu spowodowanego szybkim ruchem (ang. <i>motion blur</i> ) końcówki endoskopu: (a) obraz oryginalny oraz (b) obraz z dodanym efektem rozmycia przy ruchu „na wprost” . . . . .	58
3.23	Wyniki symulacji skończonej głębi ostrości: obraz oryginalny (a) oraz dodana głębia ostrości (b) . . . . .	59
4.1	Przepływ optyczny (f-i) wyznaczony metodą Lucas-Kanade dla kolejnych klatek filmu z wideoendoskopu (a-e), kierunek przepływu został zakodowany kolorem zgodnie z diagramem (j) . . . . .	64
4.2	Przepływ optyczny (c) wyznaczony metodą Lucas-Kanade dla obrazów wirtualnej bronchoskopii (a-b), przy ruchu kamery „na wprost” . . . . .	65
4.3	Schemat blokowy opracowanego modelu systemu nawigacji, wspomagającego pozycjonowanie końcówki bronchofiberoskopu (Zieliński i inni, 2007) . . . . .	66
4.4	Ruch kamery w przestrzeni 3D (translacja $t$ oraz rotacja $\alpha$ ) powodują zmianę współrzędnych ( $d_x, d_y$ ) rzutu punktu $P(X_0, Y_0, Z_0)$ na dwuwymiarowej płaszczyźnie obrazu . . . . .	67
4.5	(a) Uproszczony model tchawicy przyjęty w szybkiej metodzie śledzenia oraz (b) obraz ścian modelu tchawicy widziany z pozycji końcówki endoskopu (Twardowski i inni, 2006) . . . . .	69
4.6	Model powstawania obrazu w środowisku cylindrycznym. $R$ - promień cylindra, $f$ - długość ogniskowej, $r$ - odległość punktu na obrazie, $z$ - głębokość punktu na ścianie (Twardowski i inni, 2006) . . . . .	69
4.7	Wynik estymacji ruchu: (a-b) dwa obrazy wirtualnej bronchoskopii oraz (c) otrzymane cząstkowe wektory ruchu dla 12 fragmentów tchawicy (—), obliczony wektor ruchu (—) oraz znacznik obrotu (—) . . . . .	70
4.8	Zasada działania algorytmu do szacowania obrotu endoskopu: dla obrazów z dwóch chwil czasowych, otoczenie rozbłysku zostaje przetransformowane do biegunowego układu współrzędnych, tak otrzymane obrazy są korelowane w kierunku osi $\phi$ a uzyskana współrzędna maksimum współczynnika korelacji określa kąt obrotu . . . . .	71
4.9	Zasada działania algorytmu szacowania przesunięcia endoskopu: dla obrazów z dwóch chwil czasowych, obraz ścian drzewa oskrzelowego zostaje przetransformowany do biegunowego układu współrzędnych ( $r, \phi$ ), tak otrzymane obrazy dzielony są na fragmenty (zielone ramki), które są korelowane ze sobą w celu określenia przesunięcia ścian . . . . .	72
4.10	Współczynniki korelacji: (a) otrzymane przy wyznaczaniu obrotu (rysunek 4.8) oraz (b) otrzymane przy wyznaczaniu przesunięcia (rysunek 4.9) . . . . .	73

4.11	Wyniki testów na obrazach syntetycznych: (a) współrzędne $(X, Y, Z)$ trajektorii ruchu kamery „po śrubie”, (b) estymowane przesunięcie endoskopu w kierunku kariny (Zieliński i inni, 2007) . . . . .	77
4.12	Wyniki testów algorytmu estymacji położenia końcówki endoskopu na obrazach wirtualnych: (a) liczba iteracji pętli dopasowania obrazów przy wykorzystaniu jedynie informacji wzajemnej (IW) oraz w połączeniu z szybką estymacją ruchu (ER+IW), (b) obliczona odległość euklidesowa pomiędzy zadaną a wyznaczoną przez algorytm pozycją kamery (Zieliński i inni, 2007) . . . . .	77
4.13	Wynik testów algorytmu estymacji ruchu na obrazach rzeczywistych: (a) przykład klatki z zaznaczonymi wektorami ruchu obliczonymi szybką metodą szacowania przesunięcia (rozdział 4.3), (b) wartości funkcji korelacji obliczone dla poszczególnych fragmentów obrazu (Twardowski i inni, 2006) . . . . .	78
4.14	Wynik testów algorytmu estymacji ruchu na obrazie rzeczywistym: (a) skumulowane przesunięcie wzdłuż tchawicy, na którym zaznaczono gwiazdką (*) trzy klatki filmu o numerach 5, 50 i 68 (b), znajdujące się w podobnej lokalizacji (Twardowski i inni, 2006) . . . . .	78
4.15	Zależność czasu korelowania macierzy 2D w funkcji liczby wierszy dla algorytmu używającego biblioteki CUFFT . . . . .	85
A.1	Wzajemnie połączone obiekty tworzą strumień wizualizacji danych (Socha, 2011b)	92
A.2	Dane wejściowe i odpowiadający im model danych VTK (ang. <i>dataset</i> ) (Socha, 2011b)	97
A.3	Przykłady komórek o zerowych wymiarach topologicznych (1,2), 1D (3,4) oraz 2D (5-9), nazwy i oznaczenia na podstawie <code>vtkCell</code> (Schroeder i inni, 2004) . . . . .	98
A.4	Przykłady komórek o trójwymiarowej topologii (Schroeder i inni, 2004) . . . . .	99
A.5	Podstawowe typy danych (ang. <i>types of dataset</i> ) oraz obiekty biblioteki VTK umożliwiające ich obsługę (Schroeder i inni, 2004) . . . . .	101
A.6	Wzrost wydajności procesorów graficznych GPU w porównaniu z procesorami CPU: (a) wzrost wydajności przetwarzania danych oraz (b) wzrost maksymalnego transferu danych (NVIDIA, 2011b) . . . . .	103
A.7	Schematyczne porównanie współczesnych procesorów ogólnego przeznaczenia CPU oraz procesorów graficznych GPU (NVIDIA, 2011b) . . . . .	104
A.8	Schemat blokowy procesora graficznego pierwszej generacji (G80), główne oznaczenia: SP – skalarne rdzenie strumieniowe, MP – multiprocesor, TF – bloki filtrowania tekstur, L1 i L2 – pamięć podręczna, FB – pamięć obrazu (tzw. bufor ramki) (NVIDIA, 2011b) . . . . .	106
A.9	Schemat wykonywania programów ( <i>kernels</i> ) na wielu rdzeniach GPU (NVIDIA, 2011b)	107
A.10	Schemat możliwych dostępu do pamięci karty graficznej (opis w tekście) (NVIDIA, 2011b) . . . . .	109

## SPIS TABLIC

2.1	Porównanie skuteczności różnych metod diagnostycznych w określeniu anatomicznego zaawansowania choroby nowotworowej węzłów chłonnych (Toloza i inni, 2003)	23
3.1	Porównanie błędów średniokwadratowych dopasowania do prostych dla linii poziomych i pionowych na obrazach po korekcji (Socha i inni, 2005a)	54
4.1	Czas wykonywania kluczowych fragmentów kodu dla opracowanego systemu uruchomionego na procesorze Pentium 4HT 3,2GHz (Zieliński i inni, 2007)	79
4.2	Szacowany czas wyznaczenia pozycji endoskopu: wykonanie dopasowania obrazu wirtualnego do obrazu z endoskopu, dla systemu uruchomionego na procesorze Pentium 4HT 3,2GHz (Zieliński i inni, 2007)	79
4.3	Czas wykonywania kluczowych fragmentów kodu dla opracowanego algorytmu szacowania ruchu endoskopu, uruchomionego na procesorze Core2Quad 2,4GHz i karcie graficznej z procesorem GeForce 260	85
A.1	Porównanie podstawowych parametrów trzech architektur procesorów graficznych GPU firmy NVIDIA zgodnych z technologią CUDA	105
A.2	Wykaz zmiennych wbudowanych używanych do identyfikacji wątku w CUDA	108
A.3	Wykaz słów kluczowych stosowanych w programach CUDA	110